On the Adversarial Robustness of Gaussian Processes

Andrea Patanè
Christ Church College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Trinity 2020

Abstract

We study the robustness of Bayesian inference with Gaussian processes (GP) under adversarial attack settings. We begin by noticing how the distinction between model prediction and decision in Bayesian settings naturally leads us to give two different notions of adversarial robustness. The former, *probabilistic adversarial robustness*, is concerned with the behaviour of the posterior distribution, formally characterising its worst-case attack uncertainty. On the other hand, *adversarial robustness* is concerned with local stability of the model decision, and is strictly correlated with bounds on the predictive posterior distribution of the model.

In the first part of this thesis we show how, by relying on the Borell-TIS inequality, the computation of probabilistic adversarial robustness can be translated to the solution of a set of optimisation problems defined over the GP posterior mean, variance and specific derived quantities. In order to solve these, we develop a general framework for the lower- and upper-bounding of GP posterior parameters, which relies on interval bound propagation techniques, the computation of linear and lower upper bounding functions and the solution of linear and convex-quadratic programming problems. Employing the central limit theorem for stochastic processes, we then demonstrate how the derived bounds can also be used for the adversarial analysis of infinitely-wide deep BNN architectures.

In the second part of this thesis, we show how a suitably defined discretisation of the GP latent space can be used to convert the computation of adversarial robustness to the solution of a finite number of optimisation problem over a set of uni-dimensional Gaussian integral functions. We proceed by extending and adapting the GP optimisation framework developed in the context of probabilistic robustness to the formal solution of these integrals. We rely on the theory of branch-and-bound optimisation algorithms to formally prove that our method is guaranteed to terminate in finitely-many steps to an ϵ-exact solution of the problem, for any $\epsilon > 0$

selected a-priori. Furthermore, the method developed is anytime, in that it can be stopped at any point during its computation and still provide formal lower and upper bounds that can be used to certify the GP adversarial robustness. By carefully designing suitable prior functions, we then show how GPs provide us with competitive and state-of-the-art models for their application in affective computing. We finally rely on three datasets for affect recognition from physiological signals as a real-world testbed to analyse the scalability and the practical feasibility of the methods we have developed for the verification and interpretation of GP models, which we argue are crucial for the development of machine learning systems that have to interact with humans in clinical situations.

Acknowledgements

I first and foremost, express my gratitude towards my supervisor Marta Kwiatkowska. Needless to say, without the support she provided me for the last 4 years this thesis would have not been made possible.

Though not appearing as co-authors in any of the works described on this thesis, I would like to thank Giuseppe Nicosia and Nicola Paoletti. Your guidance before the start of my DPhil studies is what introduced me to research, taught me the research method and gave me in the first place the motivation to further undertake research studies.

I am also particularly thankful to Luca Laurenti and Matthew Wicker. Without the constant discussions we had almost every day for the last 2 years, this thesis would not look anything like it does today. You have also been invaluable friends throughout this whole period.

I should also thank all the co-authors that I had the pleasure to work with during my time at the University of Oxford: Alessandro Abate, Arno Blaas, Luca Bortolussi, Jan-Peter Calliess, Ginevra Carbone, Luca Cardelli, Desirée Colombo, Simon Eberz, Javier Fernández-Álvarez, Shadi Ghiasi, Alberto Greco, Giulio Lovisotto, Chris Xiaoxuan Lu, Ivan Martinovic, Kyriakos Polymenakos, Stephen Roberts, Marc Roeschlin, Stefano Rosa, Guido Sanguinetti, Enzo Pasquale Scilingo and Niki Trigoni. I also thank all the students from the research group I was part of in Oxford, the other researchers from the AffecTech network and the student from the AIMS CDT - I feel very lucky to have been part of these. The numerous discussions and exchanges of ideas we had together deeply shaped my research outlook and world view.

The work I carried out during my DPhil received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 722022 "AffecTech" and the European Research Council (ERC) project FUN2MODEL (grant agreement No. 834115). During my DPhil I was also supported by a Clarendon

scholarship and by the Autonomous Intelligent Machine and System CDT. In addition, some of the outcomes reported in this thesis are aligned with the EPSRC Programme Grant on Mobile Autonomy (EP/M019918/1).

I finally want to send a thank you to all my friends and especially to my family. Without your support nothing of this sort would be at all possible for me to do - grazie.

Andrea Patanè

October 2020, Oxford

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thanks to recent advances in optimisation, big data processing, computing infrastructure and half a century long combined research effort [76], Machine Learning (ML) has become a revolutionising force in how engineering and computational problems are today tackled. Modelling tasks that appeared too difficult to solve by an automatic learning systems just a decade ago are now considered standard benchmarks for the application of ML models. Particularly in supervised learning tasks, and especially in the form of deep Neural Networks (NNs), ML models now routinely achieve close to human-level recognition performance on applications that range from computer vision [88] to speech recognition [79], as well as physics modelling [118], biological analysis [33], clinical diagnosis [87], computer security [5], user localisation [171], and many others [144, 166, 201].

Despite these tremendous successes though, Artificial Intelligence (AI) systems built over ML models still learn, operate and reason in a fundamentally different way from how humans do, so that their nominal outstanding performance completely falls apart as soon as specific mathematical, statistical or modelling assumptions break down. In fact, several vulnerabilities have been recently discovered and investigated in the literature [189, 15, 17]. Arguably, among the most striking examples is that of local *adversarial attacks*: by performing small manipulations on an input point it is often possible to trick an ML model into predicting any of its output values in the immediate proximity of that point [189]. In end-to-end settings, in particular, the attacks produced are astonishing, with adversarial examples often being visually indistinguishable from the original input point.

Unsurprisingly, the discovery of such vulnerabilities has called for caution in applying these techniques in safety-critical situations [86]. For instance, it was observed that all it takes to trick a road-sign recognition NN in confounding a *turn right* sign with a *turn left* one is a single-pixel modification [206], the effect of which in the real

world can be disastrous. Similarly, it was demonstrated how medical diagnosis models [164], security identification systems [106], speech recognition systems [165], and many others [117, 102, 101], could all be easily fooled by means of simple adversarial attacks. This is especially the case in health-care applications, where human patients are directly involved. In the light of these findings, it is difficult to imagine how ML and AI systems can be widely deployed in the real world, without first having a thorough understanding of their shortcomings and providing guarantees over their behaviour in worst-case scenarios.

Adversarial examples were found to be so widespread in state-of-the-art NN models that it was recently investigated whether they are actually due to model misbehaviour or if they are instead an intrinsic property of certain ML models [96] or datasets [194]. Interestingly, models that are found to be easily fooled by adversarial attacks are often models that have an almost perfect empirical generalisation gap, that is, they behave similarly on a training and a test dataset. The simple fact that robustness against adversarial examples is not captured by standard generalisation metrics (e.g., root-mean-squared error or accuracy computed on a test set) suggests that there is something fundamentally different between adversarial examples and "natural" occurring points. Arguably, we cannot reasonably expect a purely data-driven ML model to perform correct predictions over adversarial examples, since it has never seen anything of the sort at training time, similarly to how we cannot expect a cat vs. dog classifier to be able to recognise the picture of a mouse. At most we could hope for an ideal ML model to show *uncertainty* in its predictions around them.

Bayesian techniques provide us with a principled way of embedding a-priori information into the training process of an ML model, so as to obtain an a-posteriori distribution on the test data, which takes into account the uncertainty inherent in the learning process. At prediction time, this is propagated through and taken into account by the decision-making pipeline, in an effort to rely only on trustworthy model decisions [137]. Intuitively, a well calibrated model uncertainty could provide a natural defence against model vulnerabilities, in fact Bayesian neural networks have been argued [64] and proven under certain strong conditions [26] to be robust to adversarial attacks.

However, most of the work on formal guarantees for ML models has been, prior to the publication of the work discussed in this thesis, focused on non-Bayesian models [203, 78, 175, 94, 107] and, to the best of our knowledge, there was no work directed at providing formal guarantees for the absence of adversarial perturbations in Bayesian

prediction settings. As such, it is currently difficult to know how Bayesian models behave in practice when performing predictions under adversarial settings.

Aim of the Thesis. In this thesis, we aim at deriving tools based on formal methods to analyse the robustness of Bayesian models under adversarial perturbation settings. Specifically, we focus on *Gaussian Process* (GP) models, which are a particular class of stochastic processes for which all the finite-dimensional distributions are multi-variate Gaussian. Because of their many favourable mathematical properties, GPs are among the most widely employed Bayesian learning approaches [209], with applications spanning robotics [178], control systems [119], biological processes [21], affective computing [190], physics modelling [80], disease diffusion [14], chemical systems [65], computer vision [32], speech recognition [155], and many others [177, 113, 54, 195, 142]. Furthermore, thanks to the central limit convergence property of stochastic processes, a large class of probabilistic models tends to behave as GPs under specific limiting conditions, so that tools developed in the context of GPs can be used as approximations for the analysis of several other modelling frameworks. This will allow us to consider the behaviour of Bayesian Neural Networks (BNNs), i.e. NNs with distributions put over their weights and biases, under adversarial attack settings, by relying on the methods developed in the context of GPs.

The resulting methodologies will then be used for the evaluation of the adversarial robustness of GP Bayesian inference models on several benchmark datasets, and in particular three datasets for affective recognition from physiological data. We will study how to provide guarantees over the GP behaviour on these health-care related tasks, and how to obtain physiologically meaningful interpretations of the GP predictions by relying on suitably derived adversarial bounds. We finally aim at investigating in practice how the robustness of a GP is affected by training, and whether accuracy is necessarily at odds with robustness even in the Bayesian settings.

Approach. We first formalise the concept of robustness of Bayesian predictions in adversarial settings. Because of the distinction that exists in Bayesian modelling between the concepts of posterior distribution, posterior predictive distribution and model decision, we explain how various definitions of adversarial robustness can be given. We discuss their meaning in terms of plausible application scenarios, and define two notions, one that characterises the probabilistic behaviour of the posterior distribution (referred to as *probabilistic adversarial robustness*) and another that captures the robustness of the model decision (referred to as *adversarial robustness*). We

then dedicate the main chapters of this thesis to showing how *over-approximations*, i.e., pessimistic estimations, for these quantities can be computed.

In particular, we show how probabilistic adversarial robustness can be over-approximated by employing the Borell-TIS inequality on the supremum random variable of a GP. This will allow us to reformulate the over-approximation problem as the solution of a set of optimisation problems defined over the GP posterior mean, variance and a specific metric space built over the GP dynamics. We rely on techniques from interval bound propagation and linear lower- and upper-approximation analysis, in order to develop a general global-optimisation framework for Gaussian process posterior models, over a wide class of kernel functions. Finally, we build a set of linear programming problems and quadratic convex programming problems (for which exact solvers are readily available [148]) that yield a pessimistic evaluation of probabilistic adversarial robustness.

We then look at adversarial robustness and see how it can be reformulated as the solution of a finite number of optimisation problems over uni-dimensional Gaussian integrals. We will observe how the same optimisation framework developed in the context of probabilistic adversarial robustness can be extended to the settings required for the computation of adversarial robustness, and formally prove that the methods that we develop converge to the actual value. In particular, we will introduce an ϵ-exact (that is, it will provably converge in finitely many steps to an over-approximation ϵ-close to the actual value) and anytime (that is, it returns a valid over-approximation of the actual quantity at any time during its computation) method.

We then rely on the central limit theorem convergence to analyse probabilistic adversarial robustness of infinitely-wide Bayesian neural networks. In particular, this allows us to perform a formal analysis of the behaviour of the posterior variance under adversarial attack settings, which provides insights into understanding of Bayesian defence mechanisms that rely on uncertainty thresholding. We then discuss the relationship between adversarial robustness and model interpretability, and see how bounds on the posterior predictive distribution can be used to obtain quantitative interpretability measures over the predictions around a given test point.

Finally, we consider three affective recognition datasets from physiological signals. We discuss how, because of the lack of sufficiently rich datasets in the field and the need to have certified/interpretable models, affective computing serves as perfect testbed for the application of GPs. We will see how competitive and state-of-the-art classification models based on GPs can be learned for affective recognition, by

carefully designing prior functions on top of well-understood preexisting physiological mathematical models. We then extend our optimisation framework to incorporate non-trivial prior functions, and show how to compute physiologically meaningful interpretation of the GP predictions in this setting.

1.1 Contributions

We develop methods for and extensively analyse the robustness of Gaussian processes under adversarial settings, as a first approach, to the best of our knowledge, for a formal analysis of adversarial examples in Bayesian ML models. In order to do so, we first formalise the notion of adversarial robustness for Bayesian learning, which leads us to the definition of two distinct robustness measures, and then develop algorithms (with lower and upper bounds) for the evaluation of such robustness metrics. We showcase the application of such methods in a range of benchmark datasets and three affective computing tasks, obtaining empirical results which suggest the fundamentally different behaviour of Bayesian models on adversarial examples when compared to their frequentist counterparts. Specifically, the main contributions of this thesis are summarised below.

- We formalise and discuss the rationale for and the relationship between two measures of robustness for Bayesian learning models under adversarial perturbation settings (probabilistic adversarial robustness and adversarial robustness), and justify their definition in terms of properties to be verified and relevant application scenarios. We then build on the probabilistic properties of GPs to find an analytical over-approximation of the probabilistic adversarial robustness of posterior GP latent models, and develop an optimisation-based framework for the explicit calculation of the parameters necessary for its computation. The developed optimisation framework is general and applicable to a wide range of kernel functions used in practice for GP learning from data. We rely on the central limit convergence property of stochastic processes to show how the techniques developed in the context of GPs can be used for verification of infinitely-wide, deep BNNs. We experimentally analyse the properties of the computed over-approximation and compare its tightness to that of approximate statistical methods. Furthermore we employ it to provide formal analysis of the posterior variance of BNNs, in an effort to formally quantify the applicability of adversarial detection techniques that rely on uncertainty thresholding.

- We next derive methods for the computation of the adversarial robustness of GP models under canonical loss functions for both the regression and the classification case. We show how, in the classification case, bounds on the predictive posterior distribution can be computed by optimising a set of uni-dimensional Gaussian integrals over their mean and variance distributional parameters. We demonstrate how the optimisation framework developed for probabilistic robustness, can be extended to these settings and implement it in a branch-and-bound optimisation scheme. By relying on the convergence theory of branch-and-bound optimisation, we then formally prove that our method terminates in finitely-many iterations to an ϵ-exact solution, for any $\epsilon > 0$ selected a-priori, and that furthermore it yields both formal over- and under-approximations of the desired quantity at any iteration.

- Employing the developed methods, we empirically observe how the adversarial robustness of the model improves with the quality of the posterior approximation performed at training time. We then demonstrate how bounds on the predictive posterior distribution can be used to formally quantify an interpretability metric over the GP predictions. Finally, we showcase the applicability of the techniques in the context of affective recognition from physiological signals. In order to do so, we first show how state-of-the-art GP models can be learned for affective tasks, by relying on prior distributions suitably defined over the dynamics of physiological mathematical models of the analysed processes. We then apply our methodology to derive physiologically meaningful interpretations over the GP predictions, demonstrating how safety can be guaranteed over the posterior GP behaviour, which we argue is crucial for machine learning systems that have to interact with humans in clinical situations.

1.2 Thesis Outline

This thesis is organised as follows. In Chapter 2 we review related work in literature, and introduce the technical background needed for the thesis in Chapter 3. In Chapter 4 we formally define adversarial robustness for Bayesian learning and give two explicit problem formulations. The main theoretical contributions are provided in Chapters 5-6, where we derive algorithms for computing robustness for GPs under adversarial settings. In Chapter 7, we benchmark our methods on three datasets for affective recognition from physiological signals. We conclude with Chapter 8 by summarising our work and highlighting possible future directions.

1.3 Publications

Much of the work presented in this thesis has been previously published in jointly authored papers. In particular, the thesis covers References [28, 19] and [70] essentially completely, and it partially takes from References [37] and [27]. In [28], we study probabilistic robustness of GPs and infinitely-wide BNNs under adversarial settings, while in [19] we develop a branch-and-bound method for the computation of adversarial robustness of classification GPs. I collaborated on the conception of these papers, the formulation of the problems, the derivation of the results, the implementation of the techniques, and I further contributed to their numerical evaluation. In [70], we show how affective recognition can be effectively performed with GPs by relying on suitably defined prior functions built around physiologically-based mathematical models. I collaborated on the definition of priors and extension of a tool for GP learning to the physiologically-inspired settings. Reference [37] presents a systematic literature review of the applications of technology in ecological momentary assessments and intervention for depressive disorders. I collaborated on reviewing and discussing works that focused on the applications of data-driven techniques for affect recognition from physiological signals. In [27], we develop a statistical verification technique for the probabilistic analysis of adversarial robustness of Bayesian neural networks. I collaborated on the conception of the paper and the formalisation of the problem.

This thesis also partially builds on [73], which is currently under review. The latter is a significant extension of the work presented in [70], in which we consider interpretability of the resulting models from the physiological perspective, obtained by employing methods for the formal analysis of GPs. I collaborated on the definition of the problem formulation, the derivation of the methodology and the implementation of the interpretability techniques.

During my DPhil, I have also co-authored the following published papers, the results of which have not been included in this thesis. In [152] we introduce a data-driven stochastic model of the physiology of a human heart, and employ statistical verification techniques for the analysis of the interaction between the heart and a rate-adaptive pacemaker. In [51] we develop an optimisation-based method for the problem of domain-adaptation of body physical and physiological signals across various recording contexts, and employ it for the evaluation of its security against impersonation attacks. In [157] we rely on Siamese neural networks to design deep architectures that are able to automatically perform subject-specific feature calibration in the context

of affective recognition from physiological signals[1]. In [156], we extend the Siamese architecture to the setting of sleep-arousal recognition from multi-modal physiological signals, and obtain the fifth place in the 2018 Physionet/CinC challenge[2]. In [172], and its journal extension [171], we derive a probabilistic model for indoor human localisation from a wrist-worn watch embedded with an IMU sensor, by relying on approximate Bayesian inference for neural networks and hidden Markov models. In [38] the protocol used in the systematic review of [37] is described. In [163] we build on the methods that we had previously developed in [28] to compute adversarial guarantees for GPs in iterative prediction settings, and embed the bounds in a safe PILCO approach, in an effort to design provably safe controllers. In [26] we prove that overparametrised BNNs are robust against gradient-based adversarial attacks in the infinite data limit. In [207] we derive an optimisation-based methodology for the under and over approximation of probabilistic adversarial robustness for BNNs.

1.4 Source Code

For the training of GP models we rely on the GPML [168] and the GPstuff [197] MATLAB toolboxes. We implemented our techniques for the verification of GPs in MATLAB along with auxiliary functions for interfacing with those two toolboxes. Techniques for the computation of probabilistic adversarial robustness for GPs and for infinitely-wide BNNs can be found at https://github.com/andreapatane/checkGP. An implementation of the techniques for the computation of adversarial robustness and the interpretability of GPs can be found at https://github.com/andreapatane/check-GPclass. GP training in the context of physiological signals, along with an extension of our approaches for the interpretability of models with a non-trivial prior function, can be found at https://github.com/shadishadi72/GP-prior.

[1] The paper obtained the best paper award at the 2018 LOD conference https://lod2018.icas.xyz/best-paper-award/.

[2] https://archive.physionet.org/challenge/2018/.

Chapter 2

Related Works

Contents

In this chapter, we review prior work related to adversarial robustness of ML models. Since the pioneering work of Szegedy *et al.* [189] many papers on this subject have appeared in the literature. Rather than covering the full spectrum, we focus our discussion on the main seminal works on adversarial attacks and on work related to Bayesian modelling. A more in depth review of adversarial attacks and defence methods can be found in [219, 212], while verification techniques and guarantees are discussed in [93, 125]. We conclude the chapter with a review of applications of Gaussian processes and related techniques in affective recognition tasks, as these constitute the main benchmark for our methods discussed in Chapter 7.

We remark that this thesis focuses only on adversarial robustness. Robustness for ML is a broad concept, and many different variants of this notion and evaluation techniques have been explored in the literature. Examples include robustness against outliers [110]; poisoning attacks [4]; labelling errors [90]; PAC bounds [149, 9]; robustness with respect to the prior distribution [12]; distributional shifts [112]; likelihood function [6]; robust approximated inference [61]; and statistical robustness [36].

2.1 Adversarial Examples

Adversarial examples are loosely defined as inputs to machine learning models intentionally crafted by an attacker with the purpose of tricking the model into performing wrong predictions. The initial observation of this phenomenon was made by Szegedy *et al.* [189] in the context of deep neural networks used for image recognition. While employing the network gradient to derive an explainability metric, the authors of [189] noticed that by performing subtle gradient-guided manipulations to a correctly classified image it was possible to force the network to predict *any* other label on that image. Formally, given a neural network f and an input image x^*, they found be possible to compute a small manipulation ϵ such that $f(x^*) \neq f(x^* + \epsilon)$, even though the images associated to x^* and $x^* + \epsilon$ looked identical to the human eye.

Since then, adversarial examples were investigated for a number of applications and models, including: speech recognition [30], pose estimation and semantic segmentation [34]; malware detection [85]; and healthcare [83, 59]. Several approaches were developed for attacking neural networks [77, 153, 206], defending against specific attack instances [154, 83, 180], explaining the reasoning behind the existence of adversarial examples [77, 96, 78], or training techniques that would optimise both for accuracy and adversarial robustness of a network [193, 213, 179]. Biggio *et al.* [16] discussed the similarity between the current line of research on adversarial examples for machine learning models and early work on spam detection avoidance using linear classifiers [40].

What perhaps was most surprising about the discovery of adversarial examples was the fact that, even state-of-the-art neural networks that were able to achieve an almost perfect generalisation accuracy on a test set, would very quickly break down under simple adversarial attacks [77]. In fact, with hindsight, it is easy to see how commonly used generalisation error metrics provide information which is unrelated to adversarial robustness. Generalisation error metrics are frequentist in nature and thus based on statistical assumptions about the data distribution [198], which fall apart in the context of adversarial examples, as it was empirically observed in [83]. This is not to say that every out-of-distribution sample would necessarily make an adversarial example; rather it suggests that, being out-of-distribution, the concept of adversarial examples is simply not captured by test set generalisation performance, so that an independent evaluation is necessary. A straightforward extension is then that of using a test set, computing adversarial attacks for all the points included in the test set, and then computing the ratio between unsuccessful attacks and the

total number of attacks performed. This provides us with an accuracy measure in adversarial attack settings [55], for which confidence intervals can be derived [121].

In order to compute such a measure exactly, it is necessary to develop formal verification techniques, which are able to give provable guarantees for the existence of adversarial examples. Roughly speaking, we want to make sure that, given a bound on the norm of the perturbation ϵ, the network is robust to any possible attack of magnitude ϵ performed over a given test point. Formally, we consider a test point x^*, a neighbourhood $T = \{x^* + \epsilon \mid ||\epsilon|| < \gamma\}$ built around x^* for $\gamma > 0$, and want to check whether:

$$f(x^*) = f(x) \qquad \forall x \in T$$

is satisfied or not [107], or alternatively compute the largest γ such that the equation above is satisfied [94]. Adversarial attacks can be seen as providing an approximate analysis of the above check. If an adversarial attack is successful, then the check above will be guaranteed to fail. On the other hand, in the case of the attack being unsuccessful, then the equation above is undecided and can still be valid or not.

Importantly, due to the existence of adversarial examples the development of verification techniques for machine learning models is of paramount importance if those are to be applied in safety-critical real-world situations. Computations of the form of the equation above can be used to provide guarantees about the behaviour of a trained model, hence increasing trust in a model predictions [170], or can be thought of as a building block towards the training of a model with certified behaviour [78]. Interestingly, as highlighted by the way in which adversarial examples were first identified, there exists also a strong connection between adversarial examples and intrepretability/explainability of machine learning models [95, 192]. In particular, in Chapter 6 we will show how our methods for the computation of adversarial guarantees on GP models will allow us to define formal, quantitative interpretability metrics around their pointwise predictions.

2.2 Adversarial Uncertainty and Robustness of Bayesian Models

The overwhelming majority of techniques for analysing machine learning models under adversarial settings have thus far been developed for frequentist learning approaches, and especially in the context of deterministic neural networks. In this thesis, we instead focus on adversarial robustness of Bayesian models, and in particular Gaussian Processes (GPs).

Bayesian modelling provides us with a probabilistically principled way of dealing with uncertainty, which is particularly appealing when dealing with adversarial examples. Intuitively, adversarial examples are points that fundamentally break our implicit statistical assumptions about the problem, and one may expect this to be captured by a well-calibrated uncertainty. Hopefully, measures of uncertainty can then be used as a means to flag and defend against plausible adversarial attacks. The first line of work that we review is based exactly on this observation, which we refer to as *uncertainty for detection of adversarial examples* (Section 2.2.1). The approach taken by these works is hybrid, in that model training is generally performed following a frequentist approach, and adversarial examples are computed for the models using standard methods. Uncertainty is mostly evaluated after model training. The second line of work that we discuss looks at the problem using a principled Bayesian perspective, i.e. model training is performed in a Bayesian way, and adversarial attacks considered are specifically tailored for Bayesian models. We will refer to this as *adversarial attacks for Bayesian methods* (Section 2.2.2). Finally, in Section 2.2.3, we will review methods developed in the context of Bayesian modelling, which aim to provide robustness guarantees over the model output.

We stress that our focus is only on Bayesian methods, that is, when the probability comes from the uncertainty over the candidate solution model itself. Other approaches in the literature consider the behaviour of a model under an uncertain input. For example, statistical techniques posit a specific distribution in the input space in order to derive a quantitative measure of robustness for deterministic networks [202, 36]. However, this approach may not be appropriate for safety-critical applications, because these typically require a worst-case analysis and adversarial examples often occupy a negligibly small portion of the input space. Alternatively, Dvijotham *et al.* [50] consider a similar problem, i.e., that of verifying deterministic deep learning models over probabilistic inputs (e.g., VAE and GAN architectures). Even though they provide stronger probability bounds than the above mentioned statistical approaches, their method is not applicable in Bayesian settings.

2.2.1 Uncertainty for Detection of Adversarial Examples

One of the advantages of Bayesian modelling is that it provides the user with pointwise uncertainty estimates for model prediction in a given test point, which can be used, for example, in active learning settings [63], for posterior estimation [108, 141] or for detecting out-of-distribution samples [171, 172]. Intuitively, a similar approach could be used for detecting adversarial examples as well. Smith *et al.* [183] analyse

the behaviour of uncertainty as estimated by Monte Carlo dropout techniques [62] on adversarial examples computed for neural network models. As a testbed, they proceed by taking points in the input space at the interpolating line between two test points in the dataset and look at how the uncertainty changes with respect to the distance from the original point. They then compute attacks on the associated deterministic neural network, and analyse how the uncertainty estimation changes with respect to adversarial attacks. They empirically find that uncertainty estimate tends to increase for adversarial points, albeit the results are not consistent across all the observations made. Bekasov *et al.* [11] empirically analyse the problem in simple adversarial sphere settings for a generalised linear model Bayesian classifier, observing a relationship between approximate model uncertainty estimations and the presence of adversarial examples. Similarly, Daubener *et al.* [42] investigate the behaviour of uncertainty on adversarial attacks computed over networks trained for speech recognition. They find that networks trained in an (approximate) Bayesian way tend to be more robust with respect to adversarial examples, and that uncertainty can be used as a threshold against adversarial attacks with some success. In an effort to improve the uncertainty estimation quality, Li *et al.* [124] derive an approximate scheme for variational inference when using α-divergences and evaluate its effectiveness in measuring uncertainty against adversarial attacks on neural networks. Bradshaw *et al.* [24], on the other hand, use GP models to better capture the model uncertainty. In particular, they learn a hybrid neural network GP model, in an effort to combine the representational capabilities of neural networks and the uncertainty estimation of GPs. They show that their hybrid model obtains a better calibration of uncertainty and is potentially able to flag adversarial examples computed by means of standard gradient optimisation methods. A different path is taken by Feinman *et al.* [56], where rather than just relying on uncertainty estimation, they perform explicit modelling of the input data distribution, and develope a combined rejection procedure for possible adversarial attacks. Rawat *et al.* [169] train Bayesian neural networks on the MNIST dataset and then empirically evaluate the behaviour of a set of uncertainty estimation measures with respect to adversarial attacks and white noise perturbations for a range of approximate Bayesian inference techniques. Interestingly, they find that the uncertainty estimation behaves similarly for adversarial attacks and for points that were perturbed by means of white Gaussian noise, and that this correlation increases as more refined uncertainty estimation techniques are used.

All the methods mentioned above fundamentally demonstrated in practice that uncertainty provides information that can be, at least to some degree, used for iden-

tifying adversarial examples. As such, they justified the importance that Bayesian techniques can have in terms of robustness. However, all these methods were relying on Bayesian properties only at test time, while model learning and adversarial attacks were mostly computed for deterministic models. Consequently, the impressive defence results obtained by these methods were later proved to be too optimistic. The rationale behind this is intuitive. Fundamentally, rejection threshold and/or procedures that are learned for an uncertainty measure end up defining a machine learning problem that aims at discerning between adversarial examples and naturally occurring points based on uncertainty. It then suffices to attack this procedure as well so that the whole defence mechanism quickly breaks down, in a typical instance of an arms race between an attacker and a defender. Grosse *et al.* [84], for example, developed an optimisation approach to craft high-confidence low-uncertainty adversarial examples in the context of Gaussian processes. They experimentally show that it is possible to design adversarial examples that are predicted with as high confidence as desired, and at the same time minimise a given uncertainty measure computed over it. Notice that uncertainty estimation for GPs is exact, so that the fact that these can be computed could not be blamed on degeneration of approximate inference methods. They furthermore show that thus computed adversarial examples transfer consistently to deterministic networks and to uncertainty measures computed for them. This shows that it is difficult in practice to develop guarantees for techniques based on uncertainty thresholding. Furthermore, Carlini *et al.* [29] developed a technique to systematically break defences based on uncertainty measures, along with 9 other popular defence techniques. This provides the necessity of developing guarantees over model behaviour rather than empirical defence mechanisms, even in Bayesian learning settings.

2.2.2 Adversarial Attacks for Bayesian Methods

While the above discussion hinted at the robustness property of Bayesian methods, at least experimentally and appealing to intuitive reasoning, Gal *et al.* [64] found a set of sufficient assumptions to guarantee that exact Bayesian learning naturally yields models that are provably robust against adversarial attacks. Similarly, in [26] we formally show how over-parametrised BNNs (and hence also GPs) are provably robust against gradient-based adversarial attacks in the infinite data limit, theoretically confirming the observations that we previously made experimentally in [19] (which will be the skeleton of Chapter 6). By extending a set of adversarial examples developed

for deterministic networks to Bayesian settings, in that work we empirically confirm the theoretical conclusions in the finite data limit, also in the BNN case.

These theoretical observations, along with the experimental ones reviewed in the previous subsection, underpinned a research effort in adversarial attacks for Bayesian methods. Interestingly, because of the distinction that exists in Bayesian modelling between the likelihood function and the loss function, as well as that between the latent variable and the output variable, after Bayesian training we are left with a posterior distribution, a posterior predictive distribution, and a model decision. The first question that comes to mind when discussing the robustness of a Bayesian model in adversarial settings is thus which one of these should we attack.

Ye *et al.* [217] take a soft, distributional approach to the question. Instead of actually computing adversarial attacks, they proceed by assuming a distribution around each point in a training set, which can be used to capture likely adversarial attacks, and then perform full posterior inference on top of that. This exploits the probabilistic properties of Bayesian models as a way to push the model learning towards more robust solutions. However, as highlighted by the authors, the approach is parametric in the sense that a prior distribution over the input space must be assumed, and that the analysis is thus not worst-case, that is, not properly adversarial in general. With the aim of robust training, Liu *et al.* [126] modify the objective function of a variational inference approach, to also take into account the robustness of a BNN against adversarial perturbations computed by means of gradient-based attacks. While doing so, the authors rightly notice that attack methods used for deterministic neural networks cannot be straightforwardly used for BNNs, because of the probabilistic nature of the latter, and hence proposed a stochastic gradient descent based method. However, Zimmermann [221] later showed that, by instead defining the attack on a Monte Carlo estimation of the BNN predictive distribution, the adversarial training procedure proposed by Liu *et al.* [126] does not actually yield significant robustness improvements compared with normal Bayesian training, calling for caution in making conclusions on attack methods that are not specifically designed for Bayesian methods. In a similar vein, Grosse *et al.* [85] extend FGSM and other gradient-based methods to the setting of Bayesian inference with GPs. Founded on the fact that in two-class classification settings the computation of the mean is sufficient to check for mis-classification, they attack the model decision by propagating gradient computations through the a-posteriori latent mean of the GP. Though the derived method does account for the Bayesian nature of GPs, it is specific to the two-class case and, even when convergence happens, it would not ensure a worst-case analysis since the

variance is not taken into account. In [27], a follow up work to that discussed in Chapter 5, we instead take a probabilistic perspective towards model robustness, which equates to attacking the posterior distribution. Namely, realisations of the BNNs are iteratively sampled from the posterior distribution and attacked by means of standard deterministic methods. The ratio between successful and unsuccessful attack is then computed as a measure of vulnerability of the networks, along with statistical guarantees computed for these quantities. This is fundamentally different to what is discussed in the above works, as neither the marginalisation step over the latent function nor the decision making step are considered by the procedure.

2.2.3 Verification for Bayesian Models

Very few works have considered adversarial robustness of Bayesian machine learning methods by taking formal guarantees into account. To the best of our knowledge, the only such work that was developed in parallel to this thesis is by Smith *et al.* [184], who derived a technique to compute adversarial robustness bound in two-class GP classification settings. The method is tailored to the ℓ_0 neighbourhood and only considers the mean of the distribution in the latent space without taking into account the uncertainty intrinsic in the GP framework (i.e., only consider the model decision). We will tackle the adversarial robustness of GPs in Chapter 6, which will be based on the work we have presented in [19]. Differently from the approach of Smith *et al.*, our approach also considers multi-class classification, takes into account the full posterior predictive distribution, and allows for exact (up to any arbitrary $\epsilon > 0$) computation under any ℓ_p norm.

Extensions and generalisations of the works discussed in this thesis can be found in [207, 163]. Specifically, in [163] we employ the techniques developed in [28] (which will be discussed in Chapter 5) to obtain probabilistic safety guarantees for GPs in iterative prediction settings. This is achieved by iteratively propagating the probabilistic adversarial bounds through discrete time steps. By further taking into account the decisions coming from a deterministic controller interfacing with the GP, we then show how the method can be used to provide guarantees over polices learned by PILCO [43]. In [207], we take a similar optimisation approach to that discussed in Chapter 5, expect that we consider the computation of probabilistic robustness in Bayesian neural networks.

Because of their mathematical properties, GPs have been widely employed in Bayesian optimisation settings, in which guarantees that we focus on are of interest. For example, Bogunovic *et al.* [20] look at obtaining robust solution to optimisation

problems. Namely, they assume that the underlying function is bounded in a Reproducing Kernel Hilbert Space (RKHS), and rely on that to compute confidence bounds on the distance between the GP and the true function. This allows them to develop upper and lower bounds on the regret, i.e. the distance between the GP prediction and the value of the actual function, and to take that into account in the solution of the optimisation problem. Though adversarial, the guarantees provided by that work are with respect to the original underlying function (which is the reason why its RKHS boundedness needs to be assumed). On the other hand, in this thesis we focus on computing guarantees for the GP model itself.

Sadigh *et al.* [178] employ GPs for modelling cyber-physical systems under uncertain environments. They tackle the problem of synthesising a safe controller for the GP model, and identify a convex subset of the probabilistic signal temporal logic, which allows them to solve the problem formally by using a mixed integer semi-definite program reformulation for the safety specification. Building on this, Sui *et al.* [188] introduce SAFEOPT, a Bayesian optimisation algorithm that additionally guarantees that, for the optimised parameters, with high probability the resulting objective function (sampled from a GP) is greater than a given threshold. Similarly, Wachi *et al.* [199] use Gaussian process guided optimisation for safely controlling Markov decision processes, and proceed by extending the SAFEOPT computation to these settings. While formal, these approaches do not give guarantees against perturbations of the synthesised parameters. For instance, they cannot guarantee that the resulting behaviour will still be safe and close to the optimal value if parameters or input states get corrupted by noise. Furthermore, property verification is achieved in a pointwise fashion, which (as discussed in Example 5 in Chapter 4) can lead to a severe over-estimation of model robustness.

A similar property to that analysed in Chapter 5 is investigated by Laurenti *et al.* [120] in the context of continuous-time continuous-space stochastic processes. In particular, they focus on linear stochastic differential equations that yield a GP as a solution, and show that probabilistic safety can be computed formally by solving a set of optimisation problems. Jackson *et al.* [98] investigate the same problem in the context of GPs learned from data over the unknown dynamics of a discrete-time system. They then verify probabilistic properties over the GP predictions by exploiting upper and lower bounds for the a-posteriori mean and variance that were developed in [28] (which will be discussed in Chapter 5).

2.3 Affective Recognition

In Chapter 7 we will apply the methodology developed for the adversarial verification of GPs in the context of affective computing. Though the origins can be traced back to earlier work [57], the current trend of research on affective analysis and computing sprang off seminal work on arousal recognition from Electrodermal Activity (EDA) [161]. The research has since evolved toward a number of different directions, including implementation and validation of processing models of emotions [69, 132]; emotional re-appraisal and Gross' theory of emotional regulation [22]; modelling and detection of the big five personality traits [75]; implementation of affective features for virtual characters [127]; signal processing and machine learning for emotion and/or mood recognition [111]; preference learning [133] and future emotion/mood prediction [190]; assistance for clinical practitioners and patients in delivering ecological momentary assessments and interventions [91]; delivering and support for computerised treatments of affective disorders [60]; quantitative analysis of the effects of life habits and sleep quality on mental well-being [190]; design practices and assisted affect regulation [58]; and many others [161].

Affect recognition often lies at the core of the methodologies mentioned above. It is in fact of particular relevance for applications of affective computing to mental health, as most affective disorders are defined as dysfunctions of the affect regulation sphere. Because of direct involvement with human-subjects, it has been argued in the literature that model safety and interpretability are necessary prior conditions for deployment of these models in real-word situations [164]. In this section we review only the work concerned with affective recognition from physiological signals, in particular from electrodermal activity and heart rate variability signals. We refer the reader to [161] for a general introduction to affective computing.

2.3.1 Affective recognition from Physiological Sensors

In particular, affective recognition from physiological sensors, i.e. the problem of inferring a user emotional/affective state from signals recorded from one's body, is routinely performed by the processing and extraction of several features from the physiological signals, e.g. by applying statistical, frequency, time/frequency, and non-linear analyses methods [181, 45, 160]. Mathematical models have been specifically developed to explain the generative process behind specific physiological signals, as a way to uncover and make explicit the relationship that exists between the affective state of a user and his/her body signals. Examples include the *integral pulse*

frequency modulation [134] and the *point-process* model [196] for the modelling of heart-rate variability dynamics; *causal modelling* [7], and *cvxEDA* [81] for explaining EDA dynamics; as well as the *recursive penalised least squares* approach for the solution of the inverse problem posed by the EEG signal generation [214]. Compared to generic feature extraction methods, model-based techniques capture and mathematically encode domain-specific expert knowledge about the physiology of the affective modelling problem itself. As such, those techniques have been shown to be able to provide a more detailed explanation of the inherent physiological mechanism underlying the observed physiological signal, and therefore resulting in interpretable metrics that allow for clinically-relevant evaluation of the features extracted from them [7].

End-to-end learning, especially in the form of deep neural network models, has been shown to consistently outperform standard ML pipelines for affective computing, at least in the case in which a sufficient amount of labelled data is available at training time [174, 182, 53, 156, 157]. Unfortunately, over-fitting problems caused by small size of datasets and the lack of interpretability that comes with deep neural networks has thus far limited the use of these methods in practical clinical applications [145, 44]. In an effort to overcome these issues, several works have looked at techniques for extensive data augmentation [92, 157] and transfer learning or pre-trained networks [74, 182], as well as learning deep models on top of hand-crafted features [99] or making ensembles of deep and shallow models [156]. While mitigating these issues, data augmentation and transfer learning, however, do not fundamentally overcome them, and the use of hand-crafted features restricts *a-priori* the learning capabilities of deep models. On the other hand, the GP model we will train in Chapter 7, by relying on patterns automatically learned from raw data by the GP, reduces the risk of over-fitting by centring the model around the explicit solution given by a physiologically inspired approach, and is thus also able to explain its predictions in terms of physiological processes.

GP models have been applied in different forms in physiological signal analysis, e.g., for solving regression tasks [48, 35, 186, 162] or as dynamical models [66]. However, physiologically-based design of the prior distribution in the Bayesian architecture of GP models has not been fully investigated, and priors used in the literature tend to be uninformative. The authors in [211] proposed an approach for designing priors for GPs specifically tailored to capturing hemodynamics in fMRI analysis. They proceeded by investigating the use of linear time-invariant systems for the prediction of the blood oxygenation level as a prior distribution, showing that an informed GP model significantly outperforms a GP trained on uninformative priors. Similarly, the

authors in [97] proposed a pseudo-Bayesian method for the estimation of intracranial pressure, where the model likelihood is informed and adapted by physiological modelling of the problem and the prior distribution is assumed to be uniform. In Chapter 7 we build on this literature to design an approach where the posterior distribution is informed both by the peculiarities of the dataset at hand and the information embedded within mathematical physiological models.

Chapter 3

Preliminaries

Contents

In this chapter we introduce the theoretical background material that is then used throughout the thesis. We start with a brief summary of the notation we use for probability spaces, random variables and stochastic processes. In Section 3.1, we review the main properties of Gaussian processes (GPs), the key probabilistic model that is then studied in the rest of the thesis. In Section 3.2, we discuss how Bayesian learning can be performed in the framework of GPs for both regression and classification problems. We review the main inference equations, the explicit form of the posterior predictive distribution and discuss decision theory. Finally, in Section 3.3 we introduce three problems for affective recognition from physiological signals, which will be used as a testbed for the method derived in this thesis.

3.1 Gaussian Processes

A *Gaussian Process* (GP) over a real-vector space is a stochastic process such that its joint distribution over any finite vector of points is a multivariate Gaussian. In this section we first introduce Gaussian processes as a particular case of stochastic processes, and then review the main properties of GPs.

We denote *probability spaces* with tuples of the form $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is a σ-algebra for Ω and P is a probability measure over $\mathcal{F}$. Given a probability space $(\Omega, \mathcal{F}, P)$ and a measurable space $(\Omega', \mathcal{F}')$, a *random variable* is a *measurable function* $\boldsymbol{\omega} : \Omega \to \Omega'$, i.e., a function such that the counter-image of any event in $\mathcal{F}'$ is still an event in $\mathcal{F}$. When it exists, we denote with p the *probability density function* of $\boldsymbol{\omega}$ associated to the probability measure P. Intuitively, an $\mathbb{R}^m$-valued *stochastic process* (or *random field*) over a real-vector space $\mathbb{R}^d$ is a collection of random variables, one for each point of the space, that take values in $\mathbb{R}^m$. Formally, this can be defined as a measurable function of the form:

$$\boldsymbol{f} : \Omega \times \mathbb{R}^d \to \mathbb{R}^m.$$

We refer to $\mathbb{R}^d$ as the input space of the process, and to $\mathbb{R}^m$ as its output space. For simplicity of notation, given $x \in \mathbb{R}^d$ we denote with $\boldsymbol{f}(x) := \boldsymbol{f}(\cdot, x) : \Omega \to \mathbb{R}^m$ the random variable induced by the stochastic process in the input point x. Similarly, given $\omega \in \Omega$ we denote with $\boldsymbol{f}(\omega) := \boldsymbol{f}(\omega, \cdot) : \mathbb{R}^d \to \mathbb{R}^m$ the trajectory of the stochastic process that corresponds to the sample ω. For simplicity of notation and when it causes no ambiguity, we will often refer to trajectories of stochastic processes simply by using their respective non-bold letters, as in $f_1 := \boldsymbol{f}(\omega_1), f_2 := \boldsymbol{f}(\omega_2), \ldots, f_M := \boldsymbol{f}(\omega_M)$, to indicate M sampled trajectories. In this last form, a stochastic process can be interpreted as a random variable over a (specific subset) of the functions $f : \mathbb{R}^d \to \mathbb{R}^m$.

We can now introduce Gaussian processes as a particular case of stochastic processes such that their joint distribution over any finite vector of input points is a multivariate Gaussian.

Definition 1 (Gaussian process)**.** *Consider a stochastic process* $\boldsymbol{f} : \Omega \times \mathbb{R}^d \to \mathbb{R}^m$*. Consider a generic vector of input points* $\mathbf{x} = [x^{(1)}, \ldots, x^{(N)}]$*, and consider the random variable* $\boldsymbol{f}(\mathbf{x}) = [\boldsymbol{f}(x^{(1)}), \ldots, \boldsymbol{f}(x^{(N)})]$*. Then, we say that* $\boldsymbol{f}$ *is a Gaussian process if* $\boldsymbol{f}(\mathbf{x})$ *has a multivariate Gaussian distribution for every choice of* $\mathbf{x}$*.*

Crucially, the definition above, even if it defines the behaviour explicitly only over finite collections of points, can be shown to be providing a definition of a process by the Kolmogorov consistency theorem [3].[1]

3.1.1 Properties of Gaussian Processes

In the finite-dimensional case, the behaviour of a Gaussian distribution can be fully characterised by the first two moments of the distribution. Similarly, a GP, $\boldsymbol{f}$, is fully characterised by a mean function $\mu : \mathbb{R}^d \to \mathbb{R}^m$ and a covariance (or *kernel*) function $\Sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{m^2}$, which are the functional counterparts of the finite-dimensional case. This can be seen as a direct consequence of the definition of Gaussian processes. In fact, let $\mathbf{x} = [x^{(1)}, \dots, x^{(N)}]$ and $S = S_1 \times \dots \times S_N$ and consider the random variable induced by the GP on $\mathbf{x}$, $\boldsymbol{f}(\mathbf{x})$. Then, by Definition 1, we have that its distribution $F_{\boldsymbol{f}(\mathbf{x})}$ is defined as:

$$F_{\boldsymbol{f}(\mathbf{x})}(S) = \int_S \frac{\exp\left(-\frac{1}{2}(\xi-\mu)^T\Sigma^{-1}(\xi-\mu)\right)}{\sqrt{(2\pi)^{mN}|\Sigma|}} d\xi$$

where $\mu = \mathbb{E}\left[\boldsymbol{f}(x^{(1)}), \dots, \boldsymbol{f}(x^{(N)})\right]$ and Σ is the $mN \times mN$ covariance matrix with generic element $\mathbb{E}\left[\left(\boldsymbol{f}(x_i) - \mathbb{E}\left[\boldsymbol{f}(x_i)\right]\right)\left(\boldsymbol{f}(x_j) - \mathbb{E}\left[\boldsymbol{f}(x_j)\right]\right)^T\right]$. As such, the behaviour of the GP on its finite-dimensional distributions remains fully defined by the definition of a function μ that assigns a mean value to each point in the input space, and a kernel function that defines the covariance between each pair of input points.

From this observation, it then follows that a series of properties that hold for Gaussian distributions generalise straightforwardly to GPs. Below we review key properties that we leverage in the rest of the thesis.

Property 1 (Linear Operations with GPs)**.** *Let $A \in \mathbb{R}^{r\times m}$ be a matrix and $\boldsymbol{f}$ be a GP with mean μ and kernel Σ. Consider the stochastic process $\boldsymbol{g} = A \cdot \boldsymbol{f}$, then $\boldsymbol{g}$ is still a GP, whose mean and covariance functions are defined as:*

$$\mu_{\boldsymbol{g}}(x) = A\mu(x) \quad \forall x$$
$$\Sigma_{\boldsymbol{g}}(x^{(1)}, x^{(2)}) = A\Sigma(x^{(1)}, x^{(2)})A^T \quad \forall x^{(1)}, x^{(2)}.$$

[1] There are some additional theoretical caveats that need to be resolved after the application of the Kolmogorov consistency theorem. For example, we will tacitly assume, throughout the thesis, the separability of the analysed processes, which ensures that the supremum of the process is well defined. A full theoretical treatment of GPs' definition falls outside of the scope of this thesis - we refer the interested reader to Reference [2] and the first Chapter of Reference [136].

Of particular relevance in Bayesian learning settings is the GP closure with respect to the conditioning operation. Intuitively, in fact, this allows us to update our belief of the distribution over a point, given an observed value on another point in the input space.

Property 2 (Conditioning with GPs)**.** *Let $\boldsymbol{f}$ be a GP with mean function μ and kernel Σ. Consider the random variable $[\boldsymbol{f}(x^{(1)}), \boldsymbol{f}(x^{(2)})]$, which is normally distributed for every $x^{(1)}$ and $x^{(2)}$ in $\mathbb{R}^d$. Then the random variable $(\boldsymbol{f}(x^{(1)})|\boldsymbol{f}(x^{(2)}) = \bar{f})$ is Gaussian with mean $\bar{\mu}$ and covariance $\bar{\Sigma}$ defined as:*

$$\bar{\mu} = \mu(x^{(1)}) + \Sigma_{x^{(1)},x^{(2)}}\Sigma^{-1}_{x^{(2)},x^{(2)}}\left(\bar{f} - \mu(x^{(2)})\right)$$
$$\bar{\Sigma} = \Sigma_{x^{(1)},x^{(1)}} - \Sigma_{x^{(1)},x^{(2)}}\Sigma^{-1}_{x^{(2)},x^{(2)}}\Sigma_{x^{(2)},x^{(1)}}.$$

Finally, a key quantity for formal verification of probabilistic systems is that of the superemum of a stochastic process, as this gives insight into the worst-case behaviour of a system. Given a subset of the input space $T \subset \mathbb{R}^d$ and a norm $||\cdot||$, the random variable associated to the superemum of a GP $\boldsymbol{f}$ is defined as:

$$\boldsymbol{f}^T_{\text{sup}} = \sup_{x \in T} ||\boldsymbol{f}(x)||$$

One of the key properties that enables probabilistic verification of GP models is that a Gaussian-shaped upper bound can be computed for the superemum random variable.

Property 3 (Borell-TIS inequality [3])**.** *Let $\boldsymbol{f}$ be a one-dimensional GP such that $\mathbb{E}[\boldsymbol{f}^T_{\text{sup}}] < \infty$ with $T \subset \mathbb{R}^d$. Let $u > \mathbb{E}[\boldsymbol{f}^T_{\text{sup}}]$ then:*

$$P(\boldsymbol{f}^T_{\text{sup}} > u) \leq \exp\left(-\frac{\left(u - \mathbb{E}[\boldsymbol{f}^T_{\text{sup}}]\right)^2}{2\sup_{x \in T}\Sigma(x)}\right)$$

3.1.2 Kernel Functions for Gaussian Processes

In applications, a GP is defined by providing an explicit functional form for the mean and kernel functions. When performing Bayesian learning with GPs, the idea is to encode our understanding about the problem at hand in the shape of the GP prior distribution, which then gets adapted according to the data observed experimentally. Usually the mean function is learned via a parametric approach, e.g., in polynomial form. In the following, we instead review briefly some of the main kernel functions used in practice, and which will be explicitly discussed in the rest of the thesis. We give the definition for GPs over a uni-dimensional output space; these can be extended to the general case by following the techniques highlighted, e.g., in [208]. We

remark that kernel designing, especially for difficult problems or for problems where little data is available, is one of the crucial steps of GP-based modelling, and a thorough discussion of the matter would be beyond the scope of this thesis. We refer the interested reader to [49, 210].

The Squared Exponential Kernel is a smooth and flexible kernel, which assumes that the correlation between random variables induced by the GP on the input space decreases with the square of their Euclidean distance. It is defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \exp\left(-\sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})^2\right).$$

The hyper-parameters of this kernel are σ^2, which is the height-scale of the the kernel, and controls the covariance ranges, and the θ_js, which are related to the length-scale, and control how quickly the GP trajectories evolve over the input space. Because of its flexibility, the squared-exponential kernel is arguably the most used kernel function in practice, especially when no additional information about the modelled problem is available. Its generalisation is given in the form of the:

The Rational Quadratic Kernel, which is defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \left(1 + \frac{1}{2}\sum_{j=1}^{m} \theta_j \left(x_1^{(j)} - x_2^{(j)}\right)^2\right)^{-\alpha}$$

with hyper-parameters σ, α and θ_j, for $j = 1, \ldots, m$. Intuitively, this is equivalent to summing a number of squared exponential kernels with different length-scales.

The Periodic Kernel is a generalisation of the squared-exponential kernel for the case in which we know that there exists periodicity in the data that we want to capture. It is defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{m} \theta_j \sin\left(p_j (x_1^{(j)} - x_2^{(j)})\right)^2\right)$$

with hyper-parameters σ, θ_j and p_j for $j = 1, \ldots, m$.

The Matérn Kernel for half-integer values is defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 k_p \exp\left(-\sqrt{\hat{k}_p \sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})^2}\right) \sum_{l=0}^{p} k_{l,p} \sqrt{\hat{k}_p \sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})^2}^{p-l}$$

with hyper-parameters σ, θ_j, for $j = 1, \ldots, m$, and (integer valued) p, while k_p, $\hat{k}_p$ and $k_{l,p}$ are constants.

Algebra with kernel can be made so as to generate new kernels. In fact it is easy to see that a linear combination (with positive coefficients) of kernels is still a kernel, and that multiplication of two kernels still yields a valid kernel function.

In the next subsection we consider a particular case of kernel functions, coming from the deep-kernel family, which are related to neural networks.

3.1.3 Infinitely-Wide Bayesian Neural Networks as Gaussian Processes

One of the most important properties of Gaussian distributions is that of the central limit convergence; that is, a suitably normalised sum of independent and identically distributed random variables converges, in the limit, to a Gaussian distribution. This is a key concept in statistical analysis because it implies that methods developed for Gaussian distributions are applicable to many problems involving other types of distributions. A similar property can be shown to hold for stochastic processes. In fact, it can be proved that, under certain limit conditions, various probabilistic models behave like GPs, as is the case, for example, for Markov processes [21] and Bayesian neural networks [147]. In particular, in Chapter 5 we employ the central limit theorem to analyse the behaviour of wide and deep Bayesian Neural Networks (BNNs) by means of their limit Gaussian process.

Specifically, a BNN is a stochastic process of the form $\boldsymbol{f}^{\mathbf{w}} : \Omega \times \mathbb{R}^d \to \mathbb{R}^m$, roughly defined by putting a distribution over the weights and biases (here all represented by the random variable vector $\mathbf{w}$) of a neural network architecture. In particular, for a neural network of depth L and activation function σ, $\boldsymbol{f}^{\mathbf{w}}(x)$ is defined as the final output of the following set of equations:

$$\begin{aligned} \zeta_i^{(l+1)} &= \sum_{j=1}^{n_l} \mathbf{W}_{ij}^{(l)} z_j^{(l)} + \mathbf{b}_i^{(l)} \quad && i = 0, \dots, n_{l+1} \\ z_i^{(l)} &= \sigma(\zeta_i^{(l)}) && i = 0, \dots, n_l \end{aligned}$$

for $l = 1, \dots, L$, where $z^{(0)} = x$, $\mathbf{W}^{(l)}$ (which is a random variable in $\mathbb{R}^{n_l \times n_{l-1}}$) and $\mathbf{b}^{(l)}$ (which is a random variable in $\mathbb{R}^{n_l}$) are the matrix of weights and vector of biases that correspond to the lth layer of the network, n_l is the number of neurons in the lth hidden layer, and where we have that $\mathbf{w} = [\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \dots, \mathbf{W}^{(L)}, \mathbf{b}^{(L)}]$.

Assume now that $\mathbf{w}$ is a vector of independent and identically distributed Gaussian random variables with mean zero, and denote with σ_b^2 the variance associated to the network biases and σ_w^2/n_l that associated to the vector weights. Then, for every layer,

we have that $\sum_{j=1}^{n_l} \mathbf{W}_{ij}^{(l)} z_j^{(l)} + \mathbf{b}_i^{(l)}$ is a sum of independent and identically distributed random variables whose, by the central limit theorem, contribution to the overall sum becomes Gaussian (as n_l approaches infinity) with mean zero and variance $\sigma_b^2 + \sigma_w^2 \mathbb{E}\left[\sigma(\zeta^{(l)})^2\right]$. It is straightforward to see that the same line of arguments can be used for any finite vector of input points $\mathbf{x} = [x^{(1)}, \ldots, x^{(N)}]$, so that in the limit the BNN behaves as a GP (though actually formally showing that the property can be propagated through consecutive layers requires a bit of extra work, see [135]).

Property 4 (Central Limit for BNNs)**.** *Let $\{\boldsymbol{f}_i^{\mathbf{w}} : \Omega \times \mathbb{R}^d \to \mathbb{R}^m\}_{i=1,\ldots,n}$ be a family of BNNs with L layers and activation function σ. Assume a Gaussian prior over the weights and biases with mean set to zero and variance respectively set to σ_w^2/n_l and σ_b^2. Assume that the family is indexed and defined in a way such that as i grows to infinity then n_l grows to infinity for every $l = 1, \ldots, L$. Then the limiting stochastic process converges in distribution to a GP $\boldsymbol{f}$ such that $\mu(x) = 0$ for every $x \in \mathbb{R}^d$ and the kernel is defined by the following recursion:*

$$\Sigma^l_{x^{(1)},x^{(2)}} = \sigma_b^2 + \sigma_w^2 F_\sigma\left(\Sigma^{l-1}_{x^{(1)},x^{(2)}}, \Sigma^{l-1}_{x^{(1)},x^{(1)}}, \Sigma^{l-1}_{x^{(2)},x^{(2)}}\right) \quad l = L, \ldots, 1$$
$$\Sigma^0_{x^{(1)},x^{(2)}} = \sigma_b^2 + \sigma_w^2 \frac{x^{(1)} \cdot x^{(2)}}{d}.$$

where F_σ is univocally determined by the choice of the activation function σ.

Though the form of the recursion is pretty straightforward, in general the function F_σ cannot be written down analytically for any choice of activation function, and Monte Carlo approximations need to be used. However, in the specific case of the ReLU activation function, the formula for the covariance function can be computed explicitly and we obtain:

$$\begin{aligned} F_\sigma\left(\Sigma^{l-1}_{x^{(1)},x^{(2)}}, \Sigma^{l-1}_{x^{(1)},x^{(1)}}, \Sigma^{l-1}_{x^{(2)},x^{(2)}}\right) &= \frac{1}{2\pi}\sqrt{\Sigma^{l-1}(x^{(1)}, x^{(1)})\Sigma^{l-1}(x^{(2)}, x^{(2)})} \\ &\quad \left(\sin\beta^{l-1}_{x^{(1)},x^{(2)}} + (\pi - \beta^{l-1}_{x^{(1)},x^{(2)}})\cos\beta^{l-1}_{x^{(1)},x^{(2)}}\right) \\ \beta^l_{x^{(1)},x^{(2)}} &= \cos^{-1}\left(\frac{\Sigma^l(x^{(1)}, x^{(2)})}{\sqrt{\Sigma^l(x^{(1)}, x^{(2)})\Sigma^l(x^{(2)}, x^{(2)})}}\right) \end{aligned} \tag{3.1}$$

for $l = 1, \ldots, L$. Similar explicit formulas can be found for other simple (though commonly used) activation functions [123, 150]. Furthermore, thanks to much recent work, a wide class of neural networks (not only fully-connected ones) with different activation functions and architectures has been proven to converge to GPs with specific structures [68, 150, 216].

3.2 Bayesian Learning with Gaussian Processes

We now review how learning from data can be done with Gaussian processes. In particular, in this thesis, we focused on supervised machine learning task, and follow the presentation of [167]. Hence, we proceed by considering a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)}) | x^{(i)} \in \mathbb{R}^d,\ y^{(i)} \in \mathcal{Y},\ i = 1, \dots, N\}$ for some input space $\mathbb{R}^d$ and output space $\mathcal{Y}$. We denote with $\mathbf{x} = [x^{(1)}, \dots, x^{(N)}]$ the aggregate vector of input points, and similarly with $\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]$ the aggregate vector of output points. We assume that the set $\mathcal{Y}$ is equal to (a subset of) $\mathbb{R}^m$ in case of regression problems, and to the discrete set $\{1, \dots, m\}$ in the case of an m-class classification problem.

In supervised learning, we are interested in finding a function $\boldsymbol{f}(x)$ that has likely generated the data contained in the dataset $\mathcal{D}$, so that given an unseen input x^* we are able to estimate its associated output value by inspecting the prediction $\boldsymbol{f}(x^*)$. Bayesian modelling approaches the learning problem with a probabilistic prospective. Specifically, it is used to keep track and to update our belief with respect to the problem solution and the likelihood of the observed data. The two ingredients for learning are then a *prior distribution* over the candidate solution functions, $p(f(x))$, which defines a (prior) stochastic process $\boldsymbol{f}$ over the solution space, and a noise model (or likelihood), which captures the probabilistic model according to which the inputs generate the outputs, $p(y|f(x))$.[2] The dataset is hence used to update our belief about the problem solution, in terms of the posterior computation according to the Bayes' rule:[3]

$$p(f(\mathbf{x})|\mathcal{D}) = \frac{p(\mathbf{y}|f(\mathbf{x}))p(f(\mathbf{x}))}{p(\mathcal{D})}, \tag{3.2}$$

where $p(\mathcal{D}) = \int p(\mathbf{y}|f(\mathbf{x}))p(f(\mathbf{x}))df(\mathbf{x})$.[4] Given a previously unseen point x^*, we then have that $p(f(x^*)|\mathcal{D})$ is the distribution of the random variable that represents our *posterior* belief about the system state in the point x^*. Indicating with y^* the variable associated to the output on the test point x^*, the posterior predictive distribution

[2] As abbreviated notation, for two random variables $\mathbf{z}_1$ and $\mathbf{z}_2$ we will denote with $p(z_1|z_2)$ the probability density function for $\mathbf{z}_1$ given the observation $\mathbf{z}_2 = z_2$.

[3] Notice that, for simplicity, we are defining the inference only w.r.t. the final dimensional distributions related to the observed dataset $\mathcal{D}$. A formal treatment of the infinite case can be found in the first Chapter of Reference [136].

[4] Notice that $\mathcal{D}$ is not a point of the probability space, thus writing it inside the probability density function represents an abuse of notation. This has to be understood as a short-hand notation for $y(\mathbf{x})$, that is, the observation of the stochastic process y (which is the observable one) on the points of the input space included in the vector of point $\mathbf{x}$.

can thus be obtained by marginalising out the unobservable variable f:

$$p(y^*|\mathcal{D}) = \int p(y^*|f(x^*))p(f(\mathbf{x})|\mathcal{D})df(\mathbf{x}). \tag{3.3}$$

In practice, e.g., when one aims at using the model in real-world applications, we are often interested in extracting from the posterior predictive distribution $p(y^*|\mathcal{D})$ a point value, $\hat{y}^*$, that satisfies specific criteria. In *Bayesian decision theory*, one proceeds by assuming a loss function, $L(\hat{y}^*, y^*)$, and minimising it with respect to the posterior distribution on the specific test point provided. Notice that in Bayesian learning the likelihood and the loss function are *independent* modelling choices and there is not necessarily a relationship between them. In fact, while the likelihood describes the observation noise, and it is hence assumed to be a property of the underlying process that we aim to model, the loss function represents the *cost* of making a specific choice given a specific, true, state of the system. In Chapter 4 we will explicitly distinguish two problem formulations for the evaluation of the adversarial robustness of Bayesian learning models, depending on whether we are interested in the robustness of the Bayesian model *itself* or if we are interested in the robustness of the decision $\hat{y}^*$ given a particular loss function $L(\hat{y}^*, y^*)$.

In Gaussian process learning, the prior stochastic process $\boldsymbol{f}$ is assumed to be a Gaussian process. This has many computational advantages, as the posterior distribution in the case of regression can be found analytically and in terms of matrix multiplications. For the classification case, though exact inference is not possible, several analytical approximations can be found and implemented in a straightforward way. The two cases are reviewed in the following two subsections.

3.2.1 Regression Problems

In regression settings the output variable is assumed to be varying in a continuous space, so that $\mathcal{Y} \subseteq \mathbb{R}^m$, and we are interested in modelling it as a continuous quantity. In this case the noise process is generally assumed to be a product of independent Gaussian distribution so that:

$$p(y|f(x)) = \prod_{j=1}^{m} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(f_j(x) - y_j)^2}{2\sigma_j^2}\right) \tag{3.4}$$

where $\sigma = [\sigma_1, \ldots, \sigma_m]$ is the vector of noise levels of each component of the process, which captures the noise intrinsic in the observation of the quantity y.

Consider now a GP prior $\boldsymbol{f}$. As the convolution between two Gaussian distributions is still Gaussian, then it follows that the posterior distribution defined by

Equation (3.2) is Gaussian. In particular, the formulas for the *a-posteriori* mean and variance can be computed by relying on the conditioning Property 2, so that the following holds.

Property 5 (Inference Equations for Regression)**.** *Let $\boldsymbol{f}$ be a GP with with mean function μ and kernel Σ. Let $\mathcal{D}$ be a dataset and consider the likelihood function from Equation* (3.4)*. Then the posterior stochastic process $\bar{\boldsymbol{f}} = \boldsymbol{f}|\mathcal{D}$ is a GP with finite-dimensional distribution defined by the mean and kernel:*

$$\bar{\mu}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \Sigma_{\mathbf{x}^*,\mathbf{x}} \left(\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I\right)^{-1} (\mathbf{y} - \mu(\mathbf{x}))$$
$$\bar{\Sigma}_{\mathbf{x}^*,\mathbf{x}^*} = \Sigma_{\mathbf{x}^*,\mathbf{x}^*} - \Sigma_{\mathbf{x}^*,\mathbf{x}} \left(\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I\right)^{-1} \Sigma_{\mathbf{x},\mathbf{x}^*}$$

for any $\mathbf{x}^* = [x^{*,(1)}, \ldots, x^{*,(M)}]$ finite collection of points in $\mathbb{R}^d$.

Notice how the *a-posteriori* mean and variance can be expressed in closed form in terms of analytical operations on matrices. In Chapters 5 and 6 this simple form will allow us to propagate upper and lower bounds from the input space through the GP inference equations so as to get bounds on its output. We also remark that, since the likelihood is Gaussian in this case, then the posterior predictive distribution as computed from Equation (3.3) is still Gaussian and has the same mean as the posterior distribution and variance equal to that of the posterior distribution plus a contribution determined by the likelihood noise σ^2.

Decision Theory for Regression Generally, in the regression case, the cost function is assumed to be proportional to the distance between our guess, $\hat{y}^*$, and the actual, true value y^*, hence assuming a cost that increases as the difference between our guess and the true value increases. Commonly employed loss function include the squared loss and the absolute difference loss. The optimal decision can then be found by computing the value $\hat{y}^*$ that minimises the expected risk over our posterior distribution:

$$\hat{y}^* = \arg\min_{y \in \mathbb{R}^m} \int L(y^*, y) p(y^*|\mathcal{D}) dy^*. \tag{3.5}$$

It can be shown that whenever both $L(y^*, y)$ and $p(y^*|\mathcal{D})$ are symmetric (as is the case for distance functions and Gaussian processes) then $\hat{y}^*$ is the mean of the posterior distribution, that is, $\hat{y}^* = \bar{\mu}(x^*)$. However, this does not hold in general for asymmetric loss functions, and other solutions must be found in a case by case scenario [168, 13]. In this thesis, in particular in Chapter 6 when computing robustness of model decisions, we will focus our investigation on *canonical loss functions* for regression, that is, those which are symmetric.

Maximum Likelihood Estimation for the Hyper-Parameters One of the main properties of GP regression models is that, as can be seen from the equations in Property 5, inference can be done by simple operations with matrices - so that the posterior is computed exactly. The key to obtain a good posterior model then rests mainly on the specification of a "good" prior model. This involves the selection of a suitable prior mean, a kernel function and a set of hyper-parameters for those two. The first two choices from this pose a discrete optimisation problem for which the evidence framework was developed for the computation of an approximated solution. Hyper-parameter selection is instead generally done in the continuous space. This can be done straightforwardly by maximising the model marginal likelihood with respect to values selected for the hyper-parameters. Denote with θ the set of hyper-parameters that we are interested in estimating. Defining the marginal likelihood (also called evidence) as the integral of the likelihood times the prior marginalised over the latent function, we obtain the following expression for its logarithm:

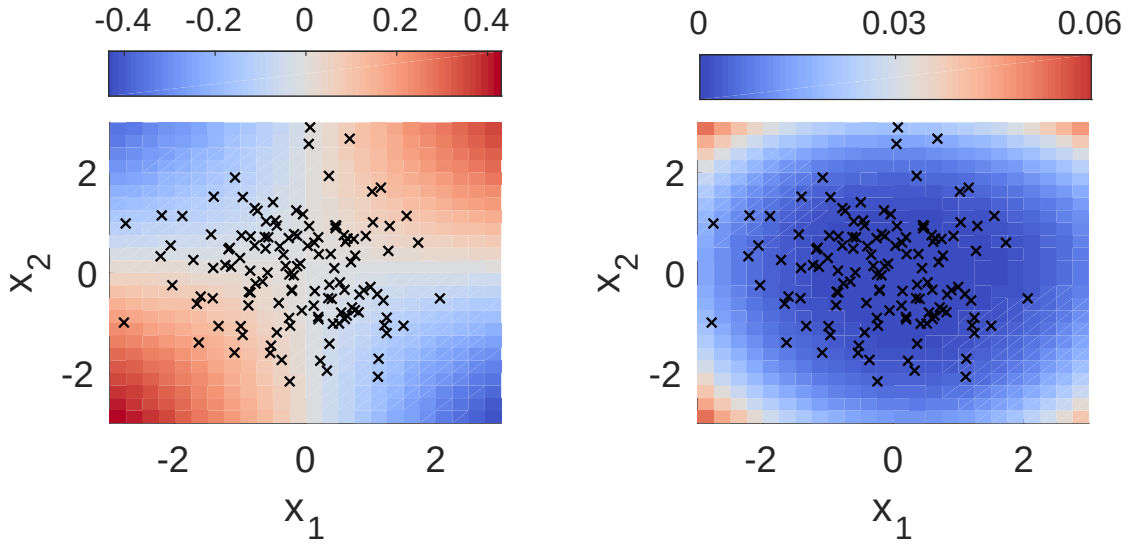
$$\log p(y(\mathbf{x})|\theta) = -\frac{1}{2}y(\mathbf{x})^T \left(\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I\right)^{-1} y(\mathbf{x}) - \frac{1}{2}\log\left|\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I\right| - \frac{N}{2}\log 2\pi.$$

The first term from the above expression is a measure of a well the GP is fitting to the data, while the second term penalises complex models. Maximisation of the marginal likelihood is a standard method in statistics used to obtain a set of hyper-parameters which balances model fit and complexity out. In the case of GP, it can be shown that the partial derivatives of the marginal likelihood with respect to θ can be computed analytically. A gradient based optimiser can be used for the maximisation of the likelihood above which provide a Maximum Likelihood Estimation (MLE) for the hyper-parameters, $\hat{\theta}$. In the remaining of this thesis, unless otherwise specified, we will rely on MLE values for all the hyper-parameters involved in the GP models used. We also remark that more involved and refined methods exist for the computation of hyper-parameters, and additional details can be found in [167].

Example 1 (GP regression model)**.** *We consider a simple two-dimensional regression problem. Given an input point $x = [x_1, x_2]$ we define its associated target as the noisy variable:*

$$y = x_1 x_2 + \epsilon$$

where for the noise level we use $\epsilon \sim \mathcal{N}(0, 0.01)$. For generating a dataset we proceed by first randomly sampling 128 points from a two-dimensional Gaussian distribution with mean zero and the 2D identity matrix as the covariance function, which yields a



(a) A-posteriori mean.

(b) A-posteriori variance.

Figure 3.1: Results of GP training on a two-dimensional regression problem. Black crosses indicate the training point locations.

vector of input points $\mathbf{x}$*. We then pass every entry of* $\mathbf{x}$ *through the target equation from above, couple them together, and standardise the output values to unity variance, so to define the training dataset,* $\mathcal{D}$*.*

To learn a GP for thus defined dataset, we set the prior mean of the GP to zero and use the squared-exponential kernel function. We estimate the hyper-parameters of the kenrel by using the MLE approach. The mean and variance of the a-posteriori GP distribution obtained after training are plotted in Figure 3.1, along with the 128 samples used for training. When using a canonical loss function we have discussed that the optimal decision is given by the GP a-posteriori mean, in fact in can be seen graphically that the a-posteriori mean closely mimic the behaviour of the quadratic function $x_1 x_2$*. For what it concerns the a-posteriori variance, we can see that this is very small around the centre of the plot. This comes from the fact that the intrinsic noise of the problem (the normalised value of* ϵ*) is tiny compared to the y-scale, and that most of the training points are centred around the plot origin. Notice how the variance slowly increases as we get farther away from the region in which training samples are, as it indicates that the GP does not have enough information to be confident in its predictions in there.*

3.2.2 Classification Problems

In the classification case, the output variable varies in a discrete set that we can think of as embedded in the set $\mathcal{Y} = \{1, \dots, m\}$, where m is the number of classes to be modelled. Similarly to what is done for generalised linear models, one proceeds by defining a continuous latent space, $\mathbb{R}^m$, and a sigmoid function $\sigma : \mathbb{R}^m \to [0,1]^m$. Intuitively, the latent variable $f \in \mathbb{R}^m$ represents our classification confidence over the m classes, and the sigmoid function performs a normalisation of the latent values so that $\pi = \sigma(f) = [\sigma_1(f), \dots, \sigma_m(f)]$ can be used as the mean parameter of a multinoulli distribution. We refer to π as the vector of class probabilities, and to its generic entry π_i as the probability of class i. In particular, in the common case in which the softmax is chosen for σ, the likelihood function for a class $i \in \{1, \dots, m\}$ is defined as:[5]

$$p(y = i|f(x)) = \sigma_i(f(x)) = \frac{\exp(f_i(x))}{\sum_{k=1}^{m} \exp(f_k(x))} \tag{3.6}$$

For classification with Gaussian processes, one proceeds by putting a GP prior $\boldsymbol{f}$ over the latent space. This induces a prior function over the classification probabilities $\boldsymbol{\pi} = \sigma(\boldsymbol{f})$. Inference on a point x^* can then be done by first computing the latent posterior distribution using Equation (3.2) with the likelihood function of Equation (3.6), by means of which we can compute the posterior latent distribution on x^* as:

$$p(f(x^*)|\mathcal{D}) = \int p(f(x^*)|f(\mathbf{x}))p(f(\mathbf{x})|\mathcal{D})df(\mathbf{x}). \tag{3.7}$$

The posterior predictive distribution is then retrieved by marginalising out the latent variable, as:

$$\pi_i(x) := p(y^* = i|\mathcal{D}) = \int \sigma_i(f(x^*))p(f(x^*)|\mathcal{D})df(x^*). \tag{3.8}$$

Unfortunately, due to the non-Gaussian nature of the likelihood function of Equation (3.6), the posterior distribution computed by means of the Bayesian formula is not Gaussian in this case (not even that over the latent space), and the posterior predictive distribution cannot be written down analytically.

Several methods have thus been developed for the approximation of the classification posterior distribution. In particular, Monte Carlo methods proceed by approximating the posterior distribution by stochastic sampling, while analytic techniques

[5]Other squash functions can be used for the choice of σ, depending on the given application and for mathematical convenience [109]. Notice that p here indicates a discrete probability distribution.

aim at giving analytical approximations of the posterior distribution. In this thesis, we are primarily interested in providing formal guarantees for Gaussian process models, and as such we focus on analytic approximations of the GP classification posterior[6]. Specifically, in Chapter 6 we show how to compute formal guarantees over the Laplace and the Expectation Propagation approximate posterior [168]. Those two approximate inference techniques are briefly summarised in the following two paragraphs. For simplicity of notation we discuss only the case of two-class classification, as that leads to many useful mathematical simplifications. In fact, when there are two classes, i.e. $\mathcal{Y} = \{1, 2\}$, it suffices to compute π_1 and then simply set $\pi_2 = 1 - \pi_1$. In this way the latent variable space can be defined to be uni-dimensional, so that $f \in \mathbb{R}$. A discussion of the general case can be found in [209].

Laplace Approximation The overall idea is to perform a Gaussian approximation of the posterior over the latent function distribution over the training set $p(f(\mathbf{x})|\mathcal{D})$ from Equation (3.7). This is done by taking a second order Taylor expansion of $p(f(\mathbf{x})|\mathcal{D})$ around its mode $\hat{\mathbf{f}}$, so that one obtains:

$$p(f(\mathbf{x})|\mathcal{D}) \approx q(f(\mathbf{x})|\mathcal{D}) = \mathcal{N}\left(\hat{\mathbf{f}}, \left(K^{-1} + W\right)^{-1}\right)$$

for some suitably defined matrices K and W.[7] $q(f(\mathbf{x})|\mathcal{D})$ can then be used to integrate out Equation (3.7). Given that the prior is a Gaussian process, then the integral can be solved at this point by simply using Property 2 and we obtain the following:

Property 6 (Inference Equations for Laplace Approximation)**.** *Given a finite collection of test points* $\mathbf{x}^*$*, the Laplace approximate latent posterior is a GP with the following mean and kernel functions:*

$$\bar{\mu}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \Sigma_{\mathbf{x}^*,\mathbf{x}} K^{-1} \hat{\mathbf{f}}$$
$$\bar{\Sigma}_{\mathbf{x}^*,\mathbf{x}^*} = \Sigma_{\mathbf{x}^*,\mathbf{x}^*} - \Sigma_{\mathbf{x}^*,\mathbf{x}} (K + W^{-1})^{-1} \Sigma_{\mathbf{x},\mathbf{x}^*}.$$

Notice how the (approximate) inference equations for the latent posterior are of the same shape as that of the regression case (see Property 5). This comes from combining the (approximate in the case of Laplace) posterior over the training set with the GP prior over the test prediction, and is hence a general property of Gaussian approximations used for classification. In fact, we will find the same shape using the

[6] In fact, since Monte Carlo methods are based on sampling, only statistical guarantees (i.e. true up to a certain statistical confidence) can be derived in those settings, see e.g. [27].

[7] Note that the conditioning in $q(f(\mathbf{x})|\mathcal{D})$ is purely symbolic and has no actual probabilistic meaning.

expectation propagation method. This will allow us to develop, in Chapter 5, a framework for the optimisation of GP posterior mean and variances that we can employ indifferently in the regression case and in the classification case (under the assumption, here, that analytic Gaussian approximations are used at inference time).

Expectation Propagation The Expectation Propagation (EP) method still works by performing a GP approximation of the posterior over the latent space; however, it achieves that by iteratively fitting a Guassian distribution by means of the EP method (which is a general technique used in statistics [143]). The overall idea is that of performing a local Gaussian approximation of the likelihood function and to perform sequential updates of the approximating parameters. While a thorough treatment of how the computation is done is outside the scope of this thesis, we state below the overall resulting shape of the inference formula, as its similarity w.r.t. that of Laplace is used in Chapter 6. Additional details can be found in [209].

Property 7 (Inference Equations for EP Approximation)**.** *Given a finite collection of test points* $\mathbf{x}^*$*, the EP approximated latent posterior is a GP with the following mean and kernel functions:*

$$\bar{\mu}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \Sigma_{\mathbf{x}^*,\mathbf{x}}(K + \tilde{\Sigma})^{-1}\tilde{\mu}$$
$$\bar{\Sigma}_{\mathbf{x}^*,\mathbf{x}^*} = \Sigma_{\mathbf{x}^*,\mathbf{x}^*} - \Sigma_{\mathbf{x}^*,\mathbf{x}}(K + \tilde{\Sigma})^{-1}\Sigma_{\mathbf{x},\mathbf{x}^*}$$

where $\tilde{\Sigma}$*,* $\tilde{\mu}$ *and* K *are computed as described in [168].*

Simplification for the Case of Probit Regression Once the posterior distribution over the latent space has been computed, this is needed for the computation of the posterior predictive distribution through the evaluation of the integral of Equation (3.8). Unfortunately, the latter is usually intractable analytically and Monte Carlo approximations or quadrature formulas have to be used. However, in the two-class case, when the *probit* likelihood function is used then the integral can be solved analytically and explicit derivation yields:

$$\pi(x^*) = \Phi\left(\frac{\bar{\mu}(x)}{\sqrt{\lambda^{-2} + \bar{\Sigma}_{x^*,x^*}}}\right)$$

where $\lambda > 0$ is the scaling factor, and Φ is the probit function. This yields an analytical formula for the predictive posterior distribution in the case of Gaussian analytical approximations, and will allow us in Chapter 6 to obtain improved formal bounds in this case.

Decision Theory for Classification The most commonly employed loss function in classification is the $0-1$ loss, which assigns a cost of 0 to correct classifications, and a cost of 1 to wrong ones. We will refer to the $0-1$ loss as the *canonical* loss function for classification problems. It can be shown that the optimal decision w.r.t. this loss can be retrieved by selecting the class associated with the index that maximises the posterior predictive distribution, that is:

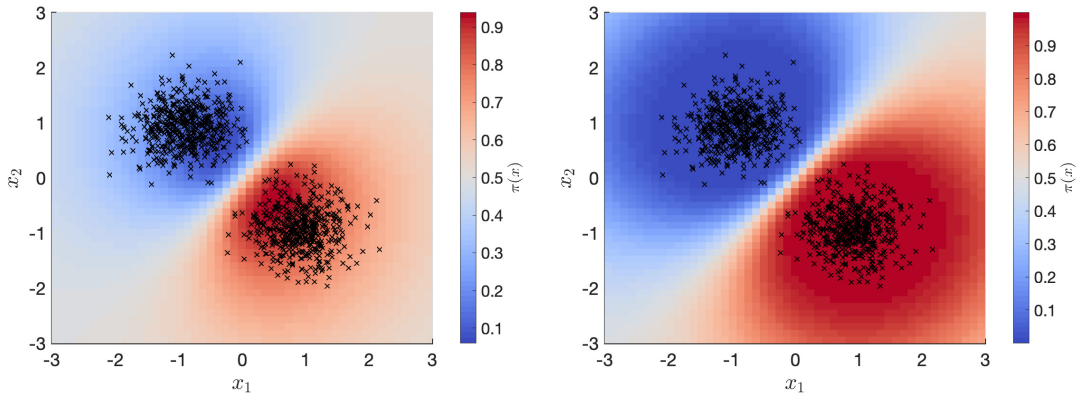
$$i^* = \arg\max_{i \in \{1,\dots,m\}} \pi_i(x^*).$$

A classifier built in such a way is also called the *optimal Bayes classifier*. Other cost functions can be used depending on the application (e.g. for medical diagnosis taking certain wrong decisions might be less "costly" than taking some other ones).

Maximum Likelihood Estimation for the Hyper-Parameters The MLE approach can be used in the classification case as well for estimating values of the hyper-parameters. In the case of the Laplace or of the EP approximated posterior distribution the form and the derivation of the formulas is similar to that of the regression case. This follows from the fact that in the Laplace and in the EP approximation we are effectively approximating the posterior over the latent function with a Gaussian process. The computation of the derivatives in these two cases is however a bit more involved, and details can be found in [168]. As for the regression models discussed in the thesis, unless otherwise specified, we employ MLE for the estimation of the hyper-parameters involved in all of the GP classification models discussed in this thesis.

Example 2 (GP classification model)**.** *We consider a simple two-class classification problem over a two-dimensional input vector $x = [x_1, x_2]$. Namely, we generate samples from a standard normal distribution and shift half of them one unit towards the bottom-right of the 2D space for the first class, and towards the top-left for the second class. We aim at learning a classification GP with prior mean set to zero and kernel function chosen as the squared-exponential function, with maximum likelihood estimation of the hyper-parameters. Since this is a two-class problem, we utilise the probit likelihood function so to obtain the simplified inference equations discussed above.*

In Figure 3.2 we compare the predictive posterior distribution obtained when using Laplace (left plot) and EP (right plot) approximation methods, along with the training samples used. Since this is a simple problem, the decision boundary obtained by the two methods is very similar, so that perfectly equal results are obtained in terms of



(a) Laplace approximated posterior.

(b) EP approximated posterior.

Figure 3.2: A two-dimensional and two-class classification problem obtained by shifting a 2-d Gaussian distribution in two difference direction.

classification accuracy. Note however how the predictive posterior distribution estimated through EP is more confident in doing prediction around the actual mode of the two Gaussian distributions, while the mode set by EP is a bit shifted too much toward the decision boundary when compared to the true value.

3.3 Affective Models

In Chapter 5 and 6 we will test the methods developed there against standard benchmark datasets for regression and classification tasks. This will allow us to investigate how they behave under controlled conditions. Then, in Chapter 7, instead, we will explore the applicability of our methodologies in real-world recognition/detection datasets on affective computing.

We will consider the problem of affective state recognition, framed as a machine learning classification problem, and Gaussian process models learned in these settings. In fact, because of the small size of the datasets typical of affective recognition tasks, deep models do not offer a viable learning paradigm in these settings. On the other hand, GPs, and Bayesian models in general, can be used to effectively exploit both the information available in a (small) training dataset and the prior information known about the problem at hand, in the form of a prior distribution assumed over the solution space, and are hence particularly appealing for this task. Furthermore, GPs provide results which, thanks to the methods that will be discussed in this thesis, are amenable to interpretation and formal analysis, which is of paramount importance for clinical applications. In particular, we focus on *arousal* and *valence* recognition

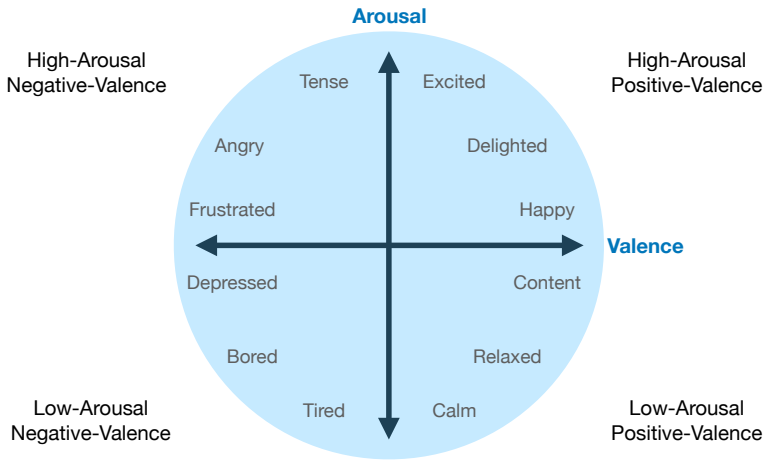


Figure 3.3: Representation of Russel's circumplex model of affect [176].

from the *Electro-Dermal Activity* (EDA) and *Heart Rate Variability* (HRV) physiological signals. That is, the input variable x of our datasets will be composed of raw, physiological signals in the form of time series of data. The output variable y will instead capture the emotion, affective and psycho-physiological state of a person. These concepts are introduced and reviewed in the rest of this section.

3.3.1 Valence and Arousal Modelling

When talking about emotion recognition, the question naturally arises of how to provide supervision for the input data. By relying on psychological theories of affect, this is usually done by assuming a model for the affective state of a person. In particular, the dimensional model of affect (also known as the Russel's circumplex model [176]) assume that the affective state of a person can be characterised by two continuous variables, denoted as *arousal* and *valence*. Intuitively, while valence codes emotional events in the positive-to-negative scale, arousal is related to the fight-or-flight response and codes our activation level as a response to the event. In Russel's model arousal and valence are constrained to a two dimensional circle, hence defining a continuous trade off between high/low arousal and high/low valence.

With classification in mind the circumplex model of affect can be discretised for emotional labels, an example of which is given in Figure 3.3. For example, boredom is described by low arousal and negative valence, while happiness may be described as positive valance and slightly high arousal situation. As such, the estimation of valence and arousal constitutes a building block toward the development of a more refined emotion recognition system. In Chapter 7 we will focus on emotion recognition and will independently analyse valence and arousal in video-induced valence recognition

tasks, and in pain-induced arousal recognition tasks. In particular, we will discretise the x-axis and the y-axis of the dimensional model of affect, so as to map it into two two-class classification problems, in which our dataset ground truth, $\mathbf{y}$, is defined according to an evaluation of the severity of the emotion induction experiment performed. More specifically we analyse the following three datasets.

CPT dataset The Cold-Pressure Test (CPT) dataset is a dataset on physically-induced arousal recognition that was introduced in [71]. It consists of EDA and HRV recordings from 26 healthy subjects. During the experiment, two diverse physiological conditions were experienced by the subjects, Condition 1, which is the resting state to ensure hemodynamic stabilization, and Condition 2, which is the presence of a physical stressor known to alter the *Autonomic Nervous System* (ANS) dynamics, i.e., cold pressure. During the resting state, the subjects were asked to sit in a comfortable position for 4 minutes while watching a black screen. During the cold-pressure test phase, instead, the subjects immersed their left hand into a tank filled with ice and water at a temperature of around 4 degrees centigrade and for a period of 3 minutes. The problem posed by the dataset is that of using the EDA and HRV signals from the subjects in order to infer whether they were recorded during the resting phase (low arousal class) or during the test phase (high arousal class). Additional details on the dataset can be found in [71].

DEAP dataset The DEAP dataset [115] consists of multi-modal physiological recordings, taken from 32 healthy subjects (19-37 years old, mean = 26.9, 50% females) while they were watching different affective video clips (40 in total). The videos were selected from a set of 120 one-minute extracts of music videos that have been previously rated by 14 volunteers for arousal and valence through an online self-assessment platform. During each trial, the index of the current trial was first shown for 2 seconds to each subject. Subsequently, 5 seconds of baseline followed. Then, the subjects were exposed to the emotional stimulus for 1 minute. Finally, they were asked to mark the mentioned dimensions on a scale of 1 to 9 after the stimulus. More details on the dataset can be found in [115]. From this dataset we extract the EDA signal, and pose the problem of inferring the valence class (low vs. high) using the latter as the only input source. In particular, we focus on the recordings associated to those videos that scored higher in terms of arousal and valence. The final dataset thus obtained consists of 105 training points.

BVHP dataset In the BVHP dataset [200], a group of 87 subjects underwent heat-induced pain experiment of four different intensities, while their physiological response was being recorded. Each pain stimulus was applied at the subject's right arm for around 5 seconds. The four temperatures were selected as those equally distributed between subject-specific pain tolerance threshold, established before the experiment itself. Each of the specific pain level stimulus was elicited 20 times in a randomised order for each study participant. There was a randomised resting phase of 8 to 12 seconds between each of the pain-inducing stimuli. Further information on the dataset can be found in [200]. From this dataset as well, we extract the EDA signal, and choose the affective state corresponding to the highest level of heat pain stimulus, and the resting state. This choice follows previous research findings that were performed on the same dataset [129, 205, 191]. As such, we obtain a two-class classification dataset made of 348 observations.

3.3.2 Electro-Dermal Activity

The term *Electro-Dermal Activity* (EDA) is used to designate variations of the electrical properties of the skin as a consequence of sweat secretion. Importantly, sweat glands are exclusively innervated by the sympathetic nervous system, so that changes in EDA directly reflect variations in the activity of the sympathetic nervous system. The latter is in turn closely related to the concept of emotional arousal, so that a direct link between EDA and arousal is believed to exist (though recent works on long term EDA monitoring challenge this explanation [159]). EDA can be measured by applying a low constant voltage to the skin, then unobtrusively measuring the resulting conductance (referred to as the skin conductance EDA signal[8]). This close link between EDA and arousal, along with the ease of measuring it, is likely the reason for the wide deployment of EDA for emotional analysis in the psychology literature, starting over 100 years ago [57].

In physiological terms, the EDA signal can be described as the super-position of two components, that is, the tonic and the phasic component [82]. These affect the overall EDA signals at different time scales, and are in turn affected by external stimuli in different ways. The tonic component is mainly responsible for the EDA baseline level, i.e. the Skin Conductance Level (SCL). On the other hand, the phasic component is constituted by the short-term responses to stimuli, also known as skin conductance responses, which are characterised by quick increase in the signal

[8]For simplicity we will refer to this simply as EDA.

amplitude, followed by an exponential decay shaped signal drop. Depending on the objective of the analysis to be carried out, approaches in the literature tend to focus either on the SCL (mid to long-term monitoring [114]) or SCR (short-term analysis for specific stimulus response [89]), or on a combination of the two.

Machine learning pipelines that build on the EDA signals for affective recognition then usually proceed by applying standard feature extraction methods to the signals. Those include statistical, frequency, time/frequency, and nonlinear analysis methods [181, 45, 160]. In Chapter 7, we will build GPs on top of a physiologically-based mathematical model of EDA, the *cvxEDA* model. Given an n samples long EDA signal x, *cvxEDA* provides a physiologically sound interpretation of its generation in terms of tonic and phasic components, in a probabilistic fashion. Namely, denoting with t the tonic component of the EDA signal, and with r its phasic component, cvxEDA models the signal generation procedure as a super-position between signals:

$$x = r + t + \epsilon \tag{3.9}$$

where ϵ is assumed to be additive white noise. The tonic activity contains information about the overall psycho-physiological state of the subject, while the phasic component shows rapid changes in EDA signals directly related to an external physiological stimulation. In physiological terms, the phasic component is the output of the convolution between the sudomotor nerve activity (SMNA) and an Impulse Response Function which describes the sweat diffusion process. We refer to the sparse SMNA driver of phasic component as p. An example of the behaviour of cvxEDA is given in Figure 3.4.

3.3.3 Heart Rate Variability

Heart Rate Variability (HRV) analysis is perhaps the most common tool used for extracting features from and analysing the human heart rate. It refers to a way of measuring how the heart rate changes with time and frequency. Its application spans many fields from emotion recognition, to evaluation of risk of vascular events for hypertensive patients [140], to evaluation of neurodegenerative processes in elderly affected by dementia [39], and many other applications in the medical settings of relevance for the quantification of cardiac or autonomic dysfunction [52, 100, 10].

The general idea behind HRV analysis for emotion recognition is that both the parasympathetic and the sympathetic nervous systems influence the heart rhythm, though relying on different signalling mechanisms and hence at different frequencies and timescales. By careful analysis of the HRV signal and its derivative, one could

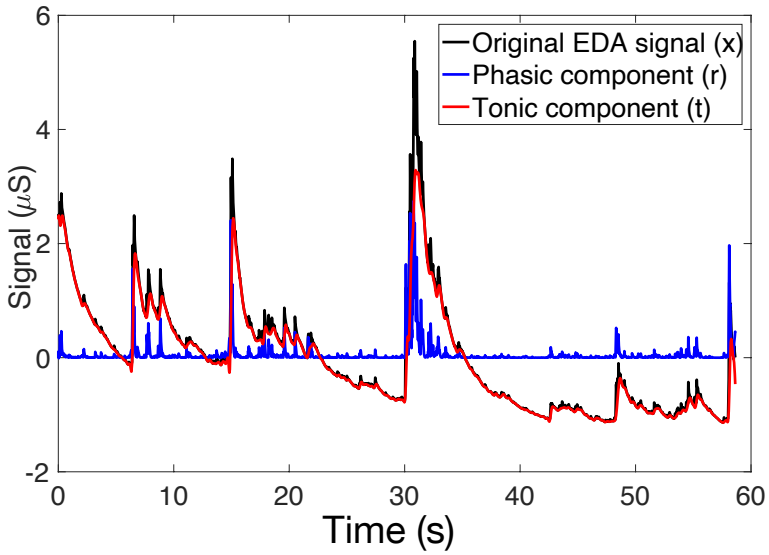


Figure 3.4: CvxEDA [81] modelling results when applied to EDA signal.

thus distinguish between parasympathetic and sympathetic activity, which are correlated to the user's arousal and valence level [1]. A number of different HRV features have been proposed in the literature. These range from simple statistical properties extracted from the HR time-signal (*time* features [151]) and geometrical characteristics of its empirical distribution (*geometric* features [1]) to sophisticated features that analyse the properties of the signal in the *frequency* domain [89], using *Poincaré* analysis [25], or that evaluate the fractal dimension and the entropy of the signal (*nonlinear* features [139]). An example of an HRV signal is shown in Figure 3.5.

3.4 Summary

In this chapter we have introduced and discussed the key concepts that form the background of this thesis. First, we have introduced Gaussian processes as a particular form of stochastic processes and reviewed their main properties. We have then discussed how GPs can be used to solve supervised machine learning problems, both in regression and classification settings, by relying on the Baysian formulation of model learning. GP models learned through Bayesian inference will be the main modelling formalism that we will employ in the rest of the thesis.

Finally, we have focused our attention on a specific setting for machine learning, affective recognition, that is, the problem of inferring from physiological signals the affective state of a subject. Because of the clinical relevance of such applications, safety is among the main concerns for practitioners in the field. As such, we will use

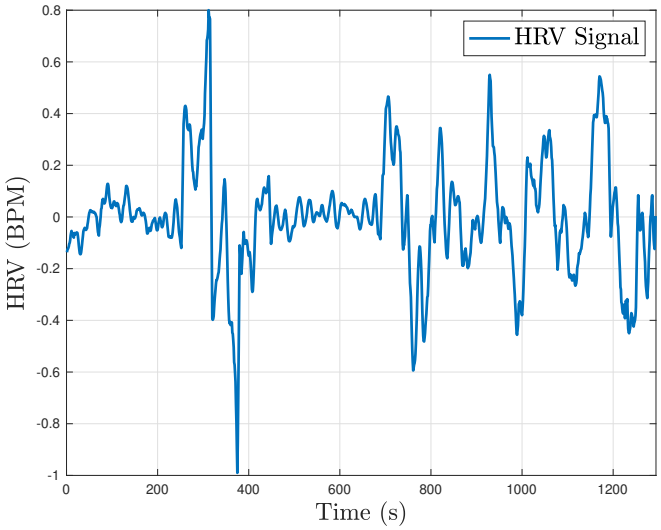


Figure 3.5: Heart rate variability signal, shown as the difference between consecutive heart rate samples.

the three affective recognition problems introduced in this chapter as a testbed for the tools that we will develop for formal verification of GP models.

In the following chapter we will formalise the notion of safety for GP models under adversarial perturbations.

Chapter 4

Robustness for Gaussian Process Models in Bayesian Inference

Contents

While robustness against adversarial attacks has a straightforward meaning in deterministic modelling, in Bayesian learning, because of the distinction between the likelihood and the loss function, training yields a posterior distribution, a posterior predictive distribution and a model decision. Thus, an obvious question that comes to mind is which one of these we should analyse the robustness of, and in which context does it make sense to do so? We have seen in the discussion of Chapter 3 that the approach so far taken in the literature is mixed, and different methods have naturally adapted themselves to slightly different definition of adversarial robustness in Bayesian settings. In this chapter we formally introduce and compare two different notions of robustness for Bayesian learning models that we then investigate in the remainder of this thesis. In particular, we focus on robustness computed in *adversarial*

settings, that is, we take a non-deterministic approach over the input space of the problem, and the measures of robustness that we compute are worst-case, in the sense that they account for the worst possible behaviour over all the input points x included in a given subset T of the input space $\mathbb{R}^d$.

In the first problem formulation (that we give in Section 4.1) we consider the robustness of the stochastic behaviour of the GP. Namely, we define worst-case guarantees over each individually sampled trajectory f from the GP posterior, and hence compute the probability (with respect to the posterior GP probability space) that sampled models are robust to adversarial attacks. We refer to this as *probabilistic adversarial robustness*. Intuitively, probabilistic adversarial robustness evaluates the uncertainty of the posterior distribution under adversarial perturbations. By deriving a simple statistical estimator for its empirical evaluation, we then discuss how probabilistic adversarial robustness is related to pointwise measures of uncertainty, and can be seen as their extension to worst-case analysis for (infinite) subsets of input points.

In this sense, probabilistic adversarial robustness is a property of the posterior model *itself*, and does not depend in any way upon the decision making procedure used on top of the Bayesian modelling. For this reason, in Section 4.2 we pose the problem of computing the *adversarial robustness* of the overall prediction-plus-decision model. Intuitively, here we are interested in providing guarantees over the non-existence of input points, coming from a given subset T, that force the optimal decision made under a given loss function to change with respect to that of a reference point $x^* \in T$. This provides us with a measure of adversarial robustness that is the direct counter-part of the adversarial robustness guarantees used in deterministic settings (as discussed in Section 2.1). In classification settings, we also discuss a quantitative version of adversarial robustness, which entails the computation of the reachability ranges over the posterior predictive distribution, which is intimately related with reachability measures used for deterministic neural networks.

We conclude the chapter with a discussion on the relationship that exists between probabilistic adversarial robustness and adversarial robustness of Bayesian models as defined here.

4.1 Probabilistic Robustness Against Adversarial Perturbations

We consider a Gaussian process $\boldsymbol{f}(x)$ defined over the input space $\mathbb{R}^d, d > 0$, with values in $\mathbb{R}^m, m > 0$, and associated probability measure P.[1]

In deterministic settings, the adversarial safety of a machine learning model f is defined as the worst-case prediction, $f(x)$, around a given input point, also known as worst-case (local) prediction to bounded adversarial perturbations [189]. Specifically, given an input point x^* one proceeds by fixing a neighbourhood T around x^* (for example a metric ball centred around it), a threshold $\delta > 0$, and checking whether

$$h_{\text{inv}}(f(x^*), f(x)) := ||f(x^*) - f(x)|| - \delta \leq 0 \qquad \forall x \in T, \tag{4.1}$$

for a given norm $||\cdot||$. We refer to the property above as δ-*invariance* and denote it with the function h_{inv}. In general, we refer to deterministic constraints as above as *specifications*. In classification settings, one is usually interested in checking whether the predicted class changes under adversarial perturbations, so that one-sided differences are more meaningful, as these capture drops in per-class classification confidence. This is encoded in the following specification:

$$h_{\text{conf}}\left(f(x^*), f(x)\right) := f_i(x^*) - f_i(x) - \delta \leq 0 \qquad \forall x \in T \tag{4.2}$$

for a given output index $i \in \{1, \ldots, m\}$. Intuitively, if h_{conf} is less than zero then one is guaranteed that the model confidence in predicting output i will not drop by more than δ in T.

In Bayesian modelling, rather than having a single model f, we are given a full distribution over the function solution space. A straightforward extension of the deterministic adversarial robustness can then be obtained by simply propagating the value of a specification h through the probability measure P associated with the stochastic process defined by the learning model. We call this probabilistic adversarial robustness, which is defined formally below.

Definition 2 (Probabilistic Adversarial Robustness)**.** *Let $T \subseteq \mathbb{R}^d$ and fix $x^* \in T$. Consider a specification function $h : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ and call*

$$\begin{aligned}\phi(x^*, T, h) &= P(\exists x \in T \; s.t. \; h\left(\boldsymbol{f}(x^*), \boldsymbol{f}(x)\right) > 0) \\ &= 1 - P(\forall x \in T \,, \; h\left(\boldsymbol{f}(x^*), \boldsymbol{f}(x)\right) \leq 0)\end{aligned}$$

[1] While the methods discussed in Chapters 5 and 6 are specific to GP models, the problem formulation discussed in this chapter applies to Bayesian learning models in general. Similar properties were in fact used in [27] and [207] for the analysis of BNNs.

Then we say that the model $\boldsymbol{f}$ is robust with probability $1-\epsilon>0$ for x^* with respect to the adversarial set T and function h iff

$$\phi(x^*, T, h) \leq \epsilon. \tag{4.3}$$

Intuitively, in Definition 2 we can consider a test point x^* and a compact set T containing x^*, and compute the probability that the realisations f of $\boldsymbol{f}$ remain close (accordingly to the transformation h) to $f(x^*)$, for each $x \in T$. We refer to

$$P_{\text{safe}}(x^*, T, h) = 1 - \phi(x^*, T, h) = P(\forall x \in T\,,\ h\left(\boldsymbol{f}(x^*), \boldsymbol{f}(x)\right) \leq 0)$$

as the safety probability, and to $\phi(x^*, T, h)$ as the probability of being unsafe. For simplicity, we assume that T is a compact set, and that both $\boldsymbol{f}$ and h are smooth, so that we can rewrite the safety probability as:

$$P_{\text{safe}}(x^*, T, h) = P(\max_{x \in T} h\left(\boldsymbol{f}(x^*), \boldsymbol{f}(x)\right) \leq 0).$$

This allows us to convert the computation of probabilistic robustness in Chapter 5 to that of the supremum of a GP, and will enable us to use the Borell-TIS inequality.

The specification h captures the concept of closeness in the output space, and its definition will, of course, depend on the particular application. The two definitions given in Equations (4.1) and (4.2), in particular, represent the case in which we are respectively interested in investigating the normed invariance of the GP output (which is of relevance, e.g., in the case in which the GP is modelling a robot that moves in a critical space), and in single-output value drops (which is of relevant, e.g., in classification problems). For simplicity, we adopt the following two notations for probabilistic adversarial safety related to h_{conf} and h_{inv}:

$$\phi_1(x^*, T, \delta) := \phi(x^*, T, h_{\text{conf}}) \tag{4.4}$$

$$\phi_2(x^*, T, \delta) := \phi(x^*, T, h_{\text{inv}}). \tag{4.5}$$

We now discuss probabilistic adversarial robustness in the two-dimensional regression problem that was introduced in Example 1.

Example 3 (Probabilistic Adversarial Robustness)**.** *Consider the GP whose training was described in Example 1. Consider now the origin point $x^o = [0, 0]$, let $\gamma = 0.1$, and define $T^o_\gamma = [x^o - \gamma, x^o + \gamma]$. As x^o is a saddle point for the target function, which is closely mimicked by the GP, variations of the mean around it are relatively small. Analogously, the variance function exhibits a flat behaviour around x^o, meaning greater confidence of the GP in performing predictions around x^o (which is a direct*

consequence of the fact that the training samples are mostly clustered around x^o and that the underlying noise has a small variance). As such, we expect realisations of the GP to be consistently stable in a neighbourhood of x^o, which in turn translates to low values for $\phi_2(x^o, T^o_\gamma, \delta)$.

On the other hand, around $x^* = [3, 3]$ the a-posteriori mean changes quickly and the variance is high, reflecting higher uncertainty, partially due to the increased distance from the region in which training samples are located. Hence, letting $T^*_\gamma = [x^* - \gamma, x^* + \gamma]$, we expect the values of $\phi_2(x^*, T^*_\gamma, \delta)$ to be greater than those computed for x^o.

In the above example we refer, of course, to the safety of the trained model. In fact, in Bayesian learning settings, the a-posteriori model is used for making predictions on unseen test points. That is, we aim at computing probabilistic adversarial robustness on the posterior stochastic process $\bar{\boldsymbol{f}} = \boldsymbol{f}|\mathcal{D}$ over the un-observable variable $\bar{f}$. We remark that this is different from the posterior predictive distribution, which is instead defined over the output variable y. Assuming that $\boldsymbol{f}$ is a GP, in the case of regression the posterior $\bar{\boldsymbol{f}}$ is still a Gaussian process, so that what was stated above actually holds independently of whether we are analysing the prior or the posterior model. As highlighted in Section 3.2.2, the situation is, in general, different for GP classification models, as the posterior distribution is not Gaussian anymore. Unless otherwise stated, in this thesis we will assume that we are working with Gaussian approximations of the GP latent posterior. Bounds on the posterior process $\pi(\bar{\boldsymbol{f}})$ over the classification probability vector can also be then computed by relying on the computation of the inverse of the employed likelihood function, σ. Our discussion will then be applicable both to the GP regression posterior and the GP classification latent posterior, and when talking about probabilistic adversarial robustness we will often refer to a GP $\boldsymbol{f}$ without mentioning whether it is a regression or a classification model. The situation is different for adversarial safety, and different notions and techniques will be introduced and derived for regression and classification models in that case.

In Chapter 5 we will derive a method for the over-approximation of adversarial probabilistic safety as defined for ϕ_1 and ϕ_2 for GPs in Bayesian learning settings. In the next subsection we derive a simple statistical estimator for ϕ that can be used to gain insights in simple, low dimensional problems.

4.1.1 Statistical Estimator for Probabilistic Adversarial Robustness

Let $f_1, \ldots, f_M$ be Monte Carlo samples from $\boldsymbol{f}$, that is, each $f_l : \mathbb{R}^d \to \mathbb{R}^m$ is a realisation of the GP $\boldsymbol{f}$. Consider now a grid of equally distanced points $x^{(1)}, \ldots, x^{(N)}$ of T, and define the quantity

$$\bar{f}_l^N = \max_{x \in \{x^{(1)}, \ldots, x^{(N)}\}} h(f_l(x^*), f_l(x)), \tag{4.6}$$

which, as N is finite, can be computed by simple enumeration. We hence define the estimator:

$$\bar{\phi}_{M,N} = \frac{1}{M} \sum_{l=1}^{M} \mathbb{I}[\bar{f}_l^N > 0] \tag{4.7}$$

where the operator $\mathbb{I}[\cdot]$ evaluates the Boolean formula in its argument to either 1 (for true) or 0 (for false). Then, assuming continuity of the sampled paths from $\boldsymbol{f}$, we have that $\bar{\phi}_{M,N}$ thus defined converges to $\phi(x^*, T, h)$ as M and N grow to infinity. To see that, it suffices to observe that from the continuity assumption we have that $\bar{f}_l^N$ converges to the actual maximum in T with probability 1 as the grid grows denser, and $\bar{\phi}_{M,N}$ is then a standard statistical estimator for a mean.

We notice that, though the estimator $\bar{\phi}_{M,N}$ will not provide formal bounds, some weak guarantees can be computed from it depending on the information available about the process $\boldsymbol{f}$. In fact, if a Lipschitz constant, L_l, is known for the sample paths f_l, it is then possible to use it for an over-approximation of the error incurred in Equation (4.6), which is, by definition of Lipschitz constant, less than $L_l D$, where D is the maximum distance between two adjacent grid points. Notice also that, for a given, fixed $N > 0$ and sequence of points $x^{(1)}, \ldots, x^{(N)}$, we have that $\bar{\phi}_{M,N}$ is a statistical estimator of the mean of a Bernoulli variable, so that confidence bounds can be easily computed for reasoning about the (statistical) quality of the approximation.

We can use this statistical estimator for the computation of the probabilistic robustness of the simple regression problem introduced above, in order to check against our interpretation of ϕ.

Example 4 (Probabilistic Adversarial Robustness - Continued)**.** *In Figure 4.1 we plot a statistical estimation of the property ϕ_2 computed in x^o and x^* with respect to the sets T_γ^o and T_γ^* (defined in Example 3) with $\gamma = 0.1$. We do this for a range of values of δ (i.e., the maximum allowed variation) ranging from 0 to 0.075.*

The values reported in the figure are estimated by using an input space grid of 2025 *equally distanced points, and by sampling* 10000 *trajectories from the posterior GP. To get an idea of the quality of the approximation provided we notice that, given that* γ *is equal to* 0.1*, the maximum distance between points in the grid is of the order of* $0.0045\sqrt{2}$*. We further notice that local Lipschitz constants of the true function can be computed in the two sets by direct inspection of the maximum of the Jacobian. Namely, we obtain* $L_{T_\gamma^o} = \sqrt{2}$ *and* $L_{T_\gamma^*} = 3.1\sqrt{2}$*. As the GP mean mimics closely the true function and the a-posteriori variance is small, for the sake of simplicity we can approximate the Lipschitz constant of the GP trajectories with that of the true function.*[2] *Hence, we have that the error that we incur because of the finite approximation of the supermum of Equation* (4.6) *is bounded by* 0.0090 *in* $L_{T_\gamma^o}$ *and by* 0.0279 *in* $L_{T_\gamma^*}$*, which in the worst-case could result in a shift to the right of the two curves reported in the plot of these two respective values. We see already that for high-dimensional and strongly non-linear systems the finite approximation would quickly become prohibitive, as a huge amount of grid points would be necessary to obtain a reasonable approximation error.*

Recall that ϕ_2 *represents the probability of finding an adversarial example in* T*, that is, a point whose distance from the prediction in the reference point is more than* δ*. Clearly, then for small values of* δ *that probability approaches* 1 *and decreases monotonically as the value of* δ *increases. It is interesting to notice that the curves follow an exponential-type of decay from the values* 1 *to* 0*. This is in accordance to the theory of GPs, where the Borell-TIS inequality (see Property 3) predicts a similar trend for the supremum random variable. Notice also how the values estimated for* x^o *are markedly smaller than those computed for* x^**, which is in accordance to the intuition provided in Example 3.*

We also can observe how the property behaves "probabilistically" with respect to δ *only in the (small) interval in which the values computed are strictly in the range* $(0, 1)$*. For example, when* $\delta > 0.06$ *we have that (statistically) the trajectories that we sample from the GP will be* δ*-invariant around* x^* *with probability 1. That also means that any decision that we can make based on the GP posterior distribution - and that follows the shape of Equation* (3.5) *- will be robust against* δ*-adversarial examples, so that probabilistic adversarial robustness implies, in certain cases, also*

[2]Notice that, given two points in the input space x and y, the joint distribution of the random variables they induce through a GP is Gaussian, so that for every $K > 0$ there is a non-null probability that they will be more than K units apart from each other. So an actual Lipschitz constant for the overall GP does not exist (a distribution of them may exist) and the argument above is approximate also in this sense.

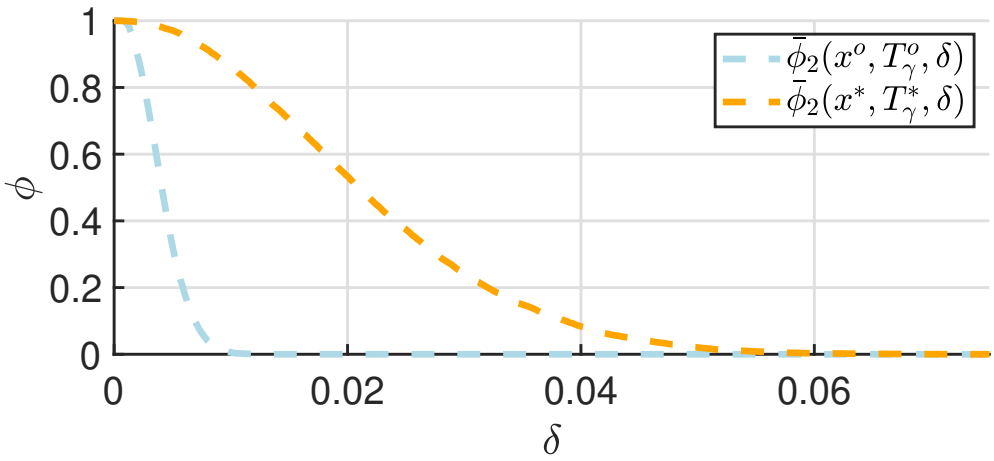


Figure 4.1: Statistical estimation of ϕ_2 in x^o and x^* with respect to the sets T^o_γ and T^*_γ for the regression problem introduced in Example 3.

guarantees on the decisions made. This observation will be formalised in Section 4.2.4, when discussing the relationship between adversarial probabilistic robustness and adversarial robustness of GPs.

4.1.2 Why Probabilistic Guarantees?

Probabilistic adversarial robustness, in a sense, considers the adversarial robustness of each model sampled from $\boldsymbol{f}$. Notice that this is different from the adversarial robustness of the GP decision (which is discussed in the next section), as it is *independent* and *prior* to the decision-making step. Hence, it is a property of the stochastic process itself, rather than the machine learning model used for making point predictions.

As such, probabilistic robustness estimations can be used, for example, to guide the decision making procedure, e.g., by adjusting the loss function used for Bayesian decision making by giving more weight to robust samples at integration time. This can be achieved by defining the loss function L used in Equation (3.5) with a weight that depends on the posterior sample chosen. Because of its probabilistic nature, probabilistic adversarial robustness can also be used for adjusting the likelihood model used at training time, and hence soft-guiding the learning toward more adversarially robust models. In fact, one can simply add to the likelihood function a term that depends on the adversarial robustness of sampled models f from $\boldsymbol{f}$. Then robustness learning follows directly from the Bayesian rule (a similar approach was taken in [126] and for deterministic neural networks in [78]).

There are cases in which the stochastic behaviour of a GP is meaningful and is not just an artefact of the learning process. For example, in control applications where a stochastic plant is modelled by a GP, then the GP posterior distribution is used in order to try to capture the uncertainty of the environment. GP realisations then describe the paths derived from the plant dynamics (that is, actual possible trajectories of the system), and safety can only be reasonably described in terms of probabilities, as there is uncertainty around the actual dynamics of the latter. In this case, probabilistic adversarial robustness can furnish guarantees for a controller that is learned under the uncertain environment.

As observed in Example 4, probabilistic adversarial robustness can be used to provide some guarantees over the possible decisions made by the model. This property will be discussed in detail in Section 4.2, where we will focus on the robustness of model decisions. Probabilistic adversarial robustness can also be seen as an extension of measures of uncertainty used in Bayesian learning. This is discussed below.

4.1.3 Probabilistic Robustness and Pointwise Uncertainty Measures

Probabilistic adversarial safety is intimately related to measures of uncertainty used in Bayesian machine learning. As an illustrative example, the author in [63] argues that aleatoric and epistemic uncertainty of a Bayesian model can be captured by looking at the *variation ratio* of its prediction on a point x^*, that is, at the quantity:

$$\frac{1}{M}\sum_{l=1}^{M}\mathbb{I}\left[h(f_l(x^*), y^*) > 0\right] \tag{4.8}$$

where f_l, $l = 1, \ldots, M$, are Monte Carlo samples from the GP, and y^* is a fixed value in the output space (e.g. the ground truth output for x^*). Note how the statistical estimator for probabilistic safety (Equation (4.7)) can be seen as a generalisation of the variation ratio in which the worst-case value from a subset of the input space T is picked and used in place of the constant value y^*. So, in a sense, probabilistic adversarial safety can be interpreted as a worst-case measure of uncertainty around an input point x^*.

Interestingly, uncertainty measures similar to that of Equation (4.8), i.e. *pointwise*, have been recently discussed as a way to flag possible adversarial examples, albeit obtaining mixed results [84, 183, 29]. Given this relationship, it is possible to build similar rejection-based defences against adversarial attacks for probabilistic adversarial robustness. In fact, we observe that pointwise measures cannot actually

provide formal guarantees on the behaviour of a stochastic process. This is because each process evolves in the input space $\mathbb{R}^d$ in a strongly correlated manner, while pointwise heuristics tacitly assume independence between the process realisations in each point of the input field. The intuition behind this is illustrated below.

Example 5 (Limitations of Pointwise Uncertainty)**.** *Consider a discrete-time stochastic process* $(\boldsymbol{f}(x), x \in \mathbb{N})$ *that takes values in* $\mathbb{R}$*, and let* $T = \{x^{(1)}, ..., x^{(10)}\}$ *be the set associated to the first 10 time instants. Assume, for simplicity, that* $\boldsymbol{f}(x^{(1)})$ *is equal to* 0 *with probability* 1*, and that we are interested in computing probabilistic safety as formulated for* h_{inv} *(defined in Equation 4.1), that is:*

$$\phi_2(x^{(1)}, T, \delta) = P(\forall x \in T,\ \boldsymbol{f}(x) < \delta),$$

for a given $\delta > 0$*. Assume that, for all* $x_i, x_j \in T$*,* $\boldsymbol{f}(x_i)$ *and* $\boldsymbol{f}(x_j)$ *are independently and equally distributed random variables such that for each* $x \in T$ *we have* $P(\boldsymbol{f}(x) < \delta) = 0.85$*. Then, if we compute the above property we obtain*

$$\phi_2(x^{(1)}, T, \delta) = 0.85^{10} \approx 0.197.$$

Thus, even though at each point $\boldsymbol{f}(x)$ *has relatively high probability of being safe,* $\phi_2(x^{(1)}, T, \delta)$ *is still small. This is because safety under adversarial perturbations depends on a set of points, and this must be accounted for to give robustness guarantees for a given stochastic model. Note that, to simplify, we used a discrete set* T*, but the same reasoning remains valid even if* $T \subseteq \mathbb{R}^d, d > 0$*, as is used in this thesis.*

In Chapter 5 we will discuss, in a case of study, the suitability of using probabilistic adversarial robustness as a defence mechanism against adversarial attacks and discuss its limitations in those settings. In the next section, we discuss adversarial robustness and compare it with probabilistic safety.

4.2 Robustness Against Adversarial Perturbations

Given a posterior GP, in Definition 2 we considered the probabilistic behaviour of the GP under adversarial perturbations. In certain applications one might be, however, more interested in the robustness of the actual decision of the GP and giving guarantees on that. When computing Bayesian optimal decisions, model decision is deterministic since the uncertainty over the output is marginalised out in the integration of Equation (3.5). As such, measures of robustness used in this setting correspond exactly with those used for deterministic learning models, the only difference being

that the output value of the GP is computed by means of the integration in Equation (3.5) for a given loss function L. Hence, we have the following.

Definition 3 (Adversarial Robustness)**.** *Let $T \subset \mathbb{R}^d$ and fix an $x^* \in T$. Consider a loss function L and a specification h. Let $y_{opt}(x)$ represent the optimal decision for a generic point x with respect to the loss function L computed for the GP $\boldsymbol{f}$. Then we say that the pair $(\boldsymbol{f}, L)$ is robust in x^* with respect to specification h and the adversarial perturbations of T, iff*

$$h(y_{opt}(x), y_{opt}(x^*)) \leq 0 \qquad \forall x \in T.$$

When discussing adversarial robustness associated with the optimal decision described in Section 3.2.1 and Section 3.2.2, we will simply refer to adversarial robustness as a property of $\boldsymbol{f}$, with the understanding that it is computed with respect to the standard loss (that is a symmetric loss for the regression case and 0-1 loss for classification). As for the case of probabilistic adversarial robustness, we assume smoothness of the specification and of the GP used for training, and that T is a compact set, so that checking for adversarial robustness can be done by the computation of the maximum of $h(y_{\text{opt}}(x), y_{\text{opt}}(x^*))$ in T.

Because of the fact that the optimal decision is different and defined over different output spaces in the cases of regression and classification, we explain in detail the notions of adversarial safety in these two cases below.

4.2.1 The Regression Case

In the regression case, given the canonical loss function, we obtain that the optimal decision is given by the mean of the posterior GP, so that the definition of adversarial safety can be made explicit. In this case we focus on invariance of the decision, that is, specification h_{inv} defined in Equation (4.1) as this is generally of interest for regression problems. We then obtain the following.

Definition 4 (Adversarial Robustness in Regression)**.** *Let $T \subset \mathbb{R}^d$ and $x^* \in T$. We say that the GP posterior regression model $\boldsymbol{f}$ is δ-robust in x^* with respect to adversarial attacks in T iff*

$$||\bar{\mu}(x^*) - \bar{\mu}(x)|| \leq \delta \qquad \forall x \in T, \tag{4.9}$$

where $\bar{\mu}$ is the a-posteriori mean of the GP.

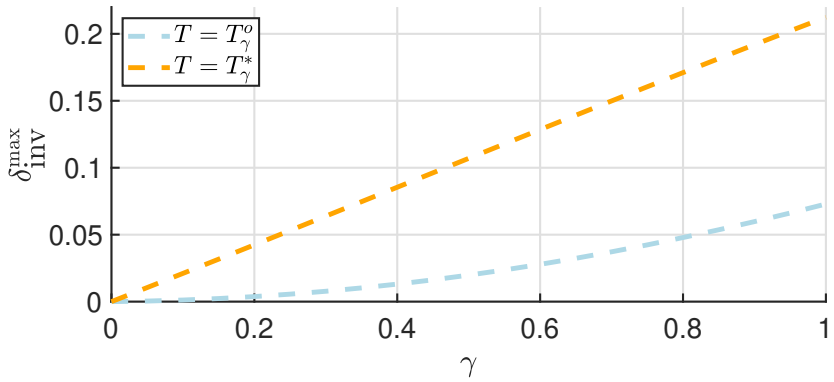


Figure 4.2: Adversarial safety threshold for the GP regression model introduced in Example 3.

The definition, is completely akin to that used in deterministic settings, for example when attacking neural networks. In fact, for a GP the mean corresponds to the maximum of the distribution so that, under convergence assumptions, it would be retrieved by a deterministic scheme that relies on regularised maximum likelihood estimations. Notice how regression adversarial safety does not take into consideration the variance of the model - it is interested only in the most likely model, among the ones obtained by Bayesian inference.

Example 6. *We evaluate adversarial safety on the GP learned on the regression task introduced in Example 3. In order to do so, we first compute* $\delta^{\max}_{inv} := \max_{x \in T} ||\bar{\mu}(x^*) - \bar{\mu}(x)||$. *Given a value* δ *then, checking whether Definition 4 is satisfied or not is equivalent to checking* $\delta^{\max}_{inv} \leq \delta$. *We do this around the point* x^o *and* x^* *defined in Example 3, and for sets* T *with the radius* γ *that varies between* 0 *and* 1. *The results of this analysis are plotted in Figure 4.2.*

For simplicity, we approximate the value of $\delta^{\max}_{inv}$ *using a grid search over* 2500 *grid points.*[3] *Similar bounds based on the Lipschitz constant as those provided in Example 4 can be computed in this case too, to get an idea of the quality of the approximation. As one would intuitively expect, in Figure 4.2, we observe that the value of* $\delta^{\max}_{inv}$ *increases as* γ *increases, as that implies larger adversarial perturbation regions* T. *Also the values computed around* x^* *are greater than those computed around* x^o, *which is in accordance with the problem intuition discussed in Example 3. It is interesting to observe that, given that the underlying true function is a quadratic function, and that the underlying model noise is small, we would expect the mean function to behave*

[3]In Chapter 5 we will see how formal bounds on the mean variation can be actually computed, so that this checking can be done in a safe manner and without sampling.

as a quadratic function. While this is the case around x^o, the behaviour around x^* is almost linear, i.e. incorrect and underestimating the true function. This follows from the fact that the training points are mostly clustered around x^o, so that around x^* the GP is actually in extrapolation regime. While this is captured very well by the model variance (plotted in Figure 3.1), adversarial robustness, relying only on the mean, cannot take that into account.

4.2.2 The Classification Case

In the classification case, given the 0-1 loss, we obtain the optimal Bayes classifier as that assigning to the input x^* the class associated to the index of the maximum of the predictive posterior distribution vector. In cases in which we are interested in checking for changes in classification, we obtain the following definition.

Definition 5 (Adversarial Robustness in Classification)**.** *Let $T \subset \mathbb{R}^d$ and fix $x^* \in T$. Consider the posterior predictive distribution π associated to the GP. Then we say that classification GP is robust in x^* with respect to adversarial attacks in T iff*

$$\arg\max_{i\in\{1,\dots,m\}} \pi_i(x) = \arg\max_{i\in\{1,\dots,m\}} \pi_i(x^*) \qquad \forall x \in T.$$

Notice that the definition for adversarial safety of the optimal Bayes classifier given above is different from its deterministic counterpart. Interestingly, adversarial safety for classification defined over the optimal Bayes classifier *does* take into account the uncertainty of the model. In fact, under convergence assumptions in the deterministic setting one may retrieve the maximum likelihood (or maximum a-posteriori) classifier, while the Bayes optimal classifier is defined by moderating the class probabilities with respect to the latent posterior distribution. Those are in general different, as the variance affects the decision made by the Bayes optimal classifier [168].

Example 7 (Adversarial Robustness)**.** *Consider the classification GP model trained with Laplace approximation that was introduced in Example 2. Since it is a two-class problem, in this case we have then that to check the estimated class of an input point it suffices to check whether its predictive posterior distribution $\pi(x)$ is greater or less then* 0.5 *- which is the decision threshold associated to the Bayes optimal classifier. The predictive posterior distribution that we obtained in Example 2 along with the training samples used are depicted again in the left plot of Figure 4.3.*

Form this, we select two reference points (orange crosses in the plot), and denote the one in the top-left of the plot as $x^{*,(1)}$, and the one close to the decision threshold

as $x^{*,(2)}$. We evaluate the adversarial safety of the two points over a neighbourhood box T_γ of radius γ, with γ that varies between 0 and 1. In order to do that, we first notice that both $x^{*,(1)}$ and $x^{*,(2)}$ are assigned to class 2 by the GP. Hence, to check for classification changes we need to check whether there exists in each one of the T_γ an x such that $\pi(x)$ is greater than 0.5. Hence, it suffices, to compute $\pi_{\max} = \max_{x \in T_\gamma} \pi(x)$ and check it against the decision threshold. For simplicity, we approximate the value of $\pi_{\max}$ with a grid search similar to the one performed in Example 6.[4]

The results for this analysis are depicted in the right plot of Figure 4.3. Of course, we obtain that $\pi_{\max}$ is monotonically increasing with the value of γ, as greater values for γ imply larger boxes T. Also notice that we do not observe any classification changes for $x^{*,(1)}$ (i.e., the pink line is always below the decision threshold 0.5). As it can be seen by inspecting the left plot, in fact variation of about $\gamma \approx 3$ would be required to cross the classification thereshold when starting from $x^{*,(1)}$. On the other hand, as $x^{*,(2)}$ is close to the classification threshold, already for $\gamma \approx 0.5$ we observe that the point is not robust to adversarial perturbations.

It is interesting to note from this simple example how the initial confidence that we put in the class of a point is not related to its robustness against adversarial perturbations. In fact, $x^{*,(2)}$ predictive distribution changes quickly from ≈ 0.05 to ≈ 0.95, while the predictive distribution around $x^{*,(1)}$ is mostly stable around 0.4. This can be seen as an illustrative example for a classification problem of what we already observed in Example 5, that is, that pointwise measures of uncertainty might be inadequate under adversarial settings.

From the example above we have seen how the computation of adversarial safety requires us to first compute bounds on the posterior predictive distribution. Those can then be checked against the decision threshold for changes in the classification output.[5] This leads us to the following generalisation of adversarial safety, which also provides a quantitative version of Definition 5.

Definition 6 (Adversarial Prediction Ranges)**.** *Let $T \subset \mathbb{R}^d$. Consider the posterior predictive distribution π associated to a GP. Let:*

$$\pi_{\min,i}(T) = \min_{x \in T} \pi_i(x) \qquad \text{for} \quad i = 1, \ldots, m$$
$$\pi_{\max,i}(T) = \max_{x \in T} \pi_i(x) \qquad \text{for} \quad i = 1, \ldots, m$$

[4] In Chapter 6 we will develop a branch-and-bound method that provably converges to $\pi_{\max}$ in finite time.

[5] Actually, in the two-class classification case, assuming a decision bound of 0.5, it suffices to check the mean of the posterior distribution. However, on other occasions (i.e., multi-class classification or thresholds different from 0.5) we have to evaluate the full posterior predictive distribution [209].

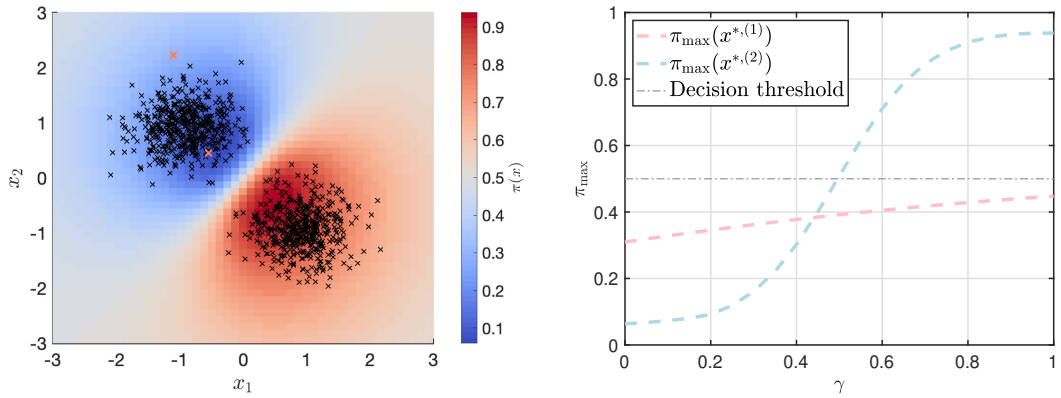


Figure 4.3: **Left**: A two-dimensional and two-class classification problem obtained by shifting a 2-d Gaussian distribution in two difference direction. **Right**: Grid estimation of upper bounds on the posterior predictive estimations in two points from the dataset. The bounds can be used to check for adversarial robustness by checking whether they cross, or not, the decision threshold line (drawn at 0.5).

then we call adversarial prediction range in T for class $i \in \{1, \ldots, m\}$ the value:

$$\delta_i(T) = \pi_{\max,i}(T) - \pi_{\min,i}(T).$$

It is easy to see that the computation of the adversarial prediction ranges poses a more general problem than that of adversarial robustness in the classification case, and the knowledge of all the $\pi_{\min,i}(T)$ and $\pi_{\max,i}(T)$ can be used straightforwardly to provide guarantees on the adversarial robustness of the GP classification model.

Definition 6 enables a similar quantitative measure to that computed for deterministic neural networks in [175]. As discussed above for Definition 5, the difference from the deterministic notion stems from the fact that in the Bayesian optimal classifier we take into consideration the moderated class probabilities, and not just the maximum likelihood solution. As such, the computed classification ranges actually take into account the classification uncertainty as well.

Notice that, while the computation of the prediction ranges furnishes useful information in the case of classification model, it is completely trivial for regression problems. In fact, the predictive posterior of a GP regression model is a Gaussian distribution, so that its support is always the whole of the real line $\mathbb{R}$.

4.2.3 Why Adversarial Robustness?

Adversarial robustness takes into consideration both the posterior distribution over the GP and the decision made on top of that. In a sense it is a straightforward generalisation of the concept of test set accuracy used in frequentist learning paradigms

to evaluate the generalisation capabilities of a model, though under the assumption of worst-case adversarial perturbations. As such, adversarial robustness as defined above is mostly relevant when we are primarily interested in the point prediction made by the GP, rather than its stochastic properties, that is, when we look at the machine learning pipeline results from the GP *as a whole*. For this reason, adversarial robustness (at least in the deterministic case) is often studied along with classification problems, where the estimated class output for an unseen test point is used down the modelling pipeline as a basis for additional decisions. For example, this is the case when using machine learning models for medical diagnosis, where at the end of the day the doctor has to make a single choice over the condition of their patient.

Another range of applications for adversarial robustness is when we are using the GP to model a controller, as for example it is the case for the HCAS case of study [104] or for self-driving cars (though bare GPs are generally not employed for this problem). It makes sense in this case to enact just one decision, the optimal one, as the concept of a car that drives probabilistically is not a very appealing one. In these scenarios the practitioner is usually more interested in having guarantees about the final output of the model, rather than the stochastic behaviour of its unobservables.

4.2.4 Adversarial Robustness and Probabilistic Robustness

We now consider the mutual relationship between adversarial robustness and probabilistic robustness. As discussed above, the main difference between the two notions lies in our attitude toward the uncertainty captured by the Bayesian model. While this is fully taken into account by probabilistic robustness, we marginalise it out before the computation of adversarial robustness. Fundamentally, from the modelling perspective, this comes from the fact that probabilistic robustness analyses the robustness of the GP posterior model, in the form of the posterior distribution over the unobservable variable f, while adversarial robustness looks at the final output of the modelling, that is, given by the decision making procedure which estimates the (observable) output y. In this sense, being a probabilistic quantity, probabilistic robustness can be used at modelling time, in order to change our model in terms of the likelihood or loss function. Adversarial robustness, instead, has more of a deterministic learning flavour, where we are only interested in the accuracy of our overall model, rather than its stochastic behaviour. Roughly speaking, the difference is analogous to that that exists between Bayesian measures of fitness and performance as estimated through frequentist indices such as the root-mean-squared-error or accuracy.

Of course, as observed in Example 4, having probabilistic guarantees implies some loose bound on adversarial safety as well, similarly to how the knowledge of a distribution can give us guarantees about its expected value, as noted in [207] in a related context. To see that, we focus for simplicity on a GP over a single un-observable, $f \in \mathbb{R}$, and consider a one-sided property. Let T be a neighbourhood around a point x^*, and consider a safety threshold δ, then we have that the safety probability in this case is given by:

$$P_{\text{safe}}(x^*, T, h_{\text{conf}}) = P\left(\forall x \in T,\ f(x^*) - f(x) \leq \delta\right) = 1 - P\left(\max_{x \in T} \left(f(x^*) - f(x)\right) > \delta\right)$$
$$= 1 - \mathbb{E}\left[\max_{x \in T} \mathbb{I}\left[f(x^*) - f(x) > \delta\right]\right].$$

Then by simple algebraic reasoning with probabilities we have that:

$$1 - \mathbb{E}\left[\max_{x \in T} \mathbb{I}\left[f(x^*) - f(x) > \delta\right]\right] \leq 1 - \max_{x \in T} \mathbb{E}\left[\mathbb{I}\left[f(x^*) - f(x) > \delta\right]\right]$$
$$= \min_{x \in T} \left(1 - \mathbb{E}\left[\mathbb{I}\left[f(x^*) - f(x) > \delta\right]\right]\right) = \min_{x \in T} \left(1 - P\left(\boldsymbol{f}(x^*) - \boldsymbol{f}(x) > \delta\right)\right)$$
$$= \min_{x \in T} \left(P\left(\boldsymbol{f}(x^*) - \boldsymbol{f}(x) < \delta\right)\right).$$

As a first observation, we then have that

$$P\left(\forall x \in T,\ \boldsymbol{f}(x^*) - \boldsymbol{f}(x) \leq \delta\right) \leq \min_{x \in T} \left(P\left(\boldsymbol{f}(x^*) - \boldsymbol{f}(x) < \delta\right)\right)$$

which extends to the continuous case the observation made in Example 5. By observing that $\boldsymbol{f}(x^*) - \boldsymbol{f}(x)$ is distributed according to a Gaussian distribution, we can employ the Chernoff concentration inequality to the argument of the minimum to obtain:

$$\min_{x \in T} \left(P\left(\boldsymbol{f}(x^*) - \boldsymbol{f}(x) < \delta\right)\right) \leq \min_{x \in T} \exp\left(-(\mu(x^*) - \mu(x)) + \frac{1}{2}\xi_{x^*,x} + \delta\right),$$

where $\xi_{x^*,x}$ is the variance of $\boldsymbol{f}(x^*) - \boldsymbol{f}(x)$. As the exponential is a monotonic function, we obtain that the right hand-side minimum can be computed by computing the minimum of the argument of exp. By definition of minimum we also have that

$$\min_{x \in T} \left(-(\mu(x^*) - \mu(x)) + \frac{1}{2}\xi_{x^*,x} + \delta\right) \leq \xi^* + \delta - \max_{x \in T}(\mu(x^*) - \mu(x)),$$

where we have set $\xi^* = \frac{1}{2}\xi_{x^*,x^*}$. So finally we obtain:

$$P\left(\forall x \in T,\ f(x^*) - f(x) \leq \delta\right) \leq \exp\left(\xi^* + \delta - \max_{x \in T}(\mu(x^*) - \mu(x))\right), \tag{4.10}$$

which relates probabilistic robustness (Definition 2) with adversarial robustness (Definition 4) in the case of regression GPs. Intuitively, checking for probabilistic adversarial robustness implies a worst-case bound on adversarial robustness as well. Of course, this is adjusted depending on the GP variance, as adversarial robustness does not explicitly depend upon the GP posterior variance. As such, probabilistic adversarial robustness can be considered as a tighter requirement than adversarial robustness.

To qualitatively analyse the relationship, let's assume for simplicity that the variance is small, so that we obtain:

$$\delta - \max_{x \in T}(\mu(x^*) - \mu(x)) \gtrsim \ln\left(P\left(\forall x \in T,\ f(x^*) - f(x) \leq \delta\right)\right).$$

That is, adversarial robustness scales as the logarithm of probabilistic robustness. If, for example, probabilistic robustness is equal to one, then we obtain $\delta - \max_{x \in T}(\mu(x^*) - \mu(x)) > 0$, which implies that the adversarial robustness condition is satisfied.

Similar formulas to that in Equation (4.10) can be obtained in the case in which the absolute value of the specification is considered, that is, by splitting the formula in two. The equations complicate somewhat for the quantitative analysis of classification, but qualitative analysis can be done by working just with the latent process $\boldsymbol{f}$ and analysing per-class changes by using the one-sided specification as done above, so that, for example, in the two-class case we might just be interested in checking for $\mu(x) > 0.5$, as that is the classification threshold of the Bayes optimal classifier.

4.3 Summary

In this chapter we considered two different notions of robustness for GPs under adversarial perturbations. Namely, we have defined probabilistic adversarial robustness, which takes into account the uncertainty of the GP posterior model, and adversarial robustness, which is computed by first marginalising out the uncertainty and then computing the optimal decision according to a given loss function. We have discussed how the two different notions are of interest for different applications and at different stages of the learning pipeline. Finally, we have discussed the quantitative relationship that exists between probabilistic adversarial robustness and adversarial robustness, by employing Chernoff's concentration inequality for Gaussian distribution. In the next chapter, we will develop a framework for the computation of probabilistic robustness of posterior GP models. The design of a branch-and-bound scheme for the computation of adversarial robustness will instead be the topic of Chapter 6.

We remark that adversarial robustness properties are not the only kind of interesting robustness properties to analyse in Bayesian settings. Depending on the assumptions, statistical robustness might be more suitable for certain purposes (e.g., under white noise corruption of the input data, as for example occurring over communication channels); as well as robustness to hyper-parameter changes or prior functions (which is particularly interesting when, as is often the case, those are unknown); or robustness against poisoning attacks of the training dataset; or over distributional shift of the test set.

Chapter 5

Probabilistic Robustness for Gaussian Processes

Contents

In this chapter we consider probabilistic robustness of Gaussian processes against adversarial perturbations in Bayesian inference settings, as defined in Section 4.1. Specifically, given a GP model trained on some dataset, a test input and a neighbourhood around the latter, we are interested in computing the probability that there exists a point in this neighbourhood such that the realisation of the GP on the latter point differs from that on the initial test input point by at least a given threshold. The measure we compute is probabilistic in the sense that it takes into account the uncertainty intrinsic in the Bayesian learning process, explicitly working with the a-posteriori distribution of the GP. In other words, probabilistic robustness is defined over the probability space induced by the GP on its output and no distribution is assumed for the input space, for which we take an adversarial perspective.

We show how an upper bound on probabilistic robustness can be computed by relying on the Borell-TIS inequality for the supremum of a GP (see Property 3 in Section 3.1). We subsequently employ *Dudley's Entropy Integral* [47], and reduce the problem of computing the GP supremum to the solution of a set of optimisation problems defined over the GP a-posteriori mean, variance and normed derivatives over a particular distance metric. Hence, we develop a general framework for the solution of these optimisation problems that relies on the computation of Lower and Upper Bounding Functions (LBFs and UBFs respectively) for the GP kernel. Building on the linearity of the GP inference equations (Properties 5, 6 and 7), we propagate LBFs and UBFs through the a-posteriori formula to obtain linear and quadratic programming problems that we solve for safe approximations of these quantities.

We first evaluate and the method discussed here on the two-dimensional regression problem introduced in Example 3 and then apply it for the analysis of the behaviour of BNNs on the MNIST handwritten digit recognition dataset. Namely, by relying on the weak convergence of infinitely-wide BNNs to GPs with deep kernels (see Property 4), we investigate the limit probabilistic robustness of fully-connected ReLU architectures to adversarial perturbations.

This chapter is organised as follows. In Section 5.1, we derive a functional form of a *safe* lower-bound to probabilistic robustness for GPs. In Section 5.2, we show how the bound constants can be computed for a general GP, under smooth assumption on the kernel function, by relying on suitably derived kernel decompositions. In Section 5.3, we analyse how to obtain fitting decompositions on the kernel functions introduced in Section 3.1.2. In Section 5.4, we discuss the computational complexity of the methodology introduced. In Section 5.5, we employ the methods for computing

probabilistic robustness of GPs on a 2-dimensional regression task and for the limit analysis of BNNs on the MNIST dataset. We report in this chapter proofs and proof sketches for the main results.

5.1 Bounding Probabilistic Safety

In this section we proceed by finding a lower bound in analytic form to probabilistic safety for a generic posterior GP model $\bar{\boldsymbol{f}} = \boldsymbol{f}|\mathcal{D}$. We first find the bound for ϕ_1 (defined in Equation (4.4)) and then generalise it to ϕ_2 (defined in Equation (4.5)) by means of the union bound of probabilities. For simplicity of presentation, we start our investigation with the assumption that the compact input set T is a box in the input space, that is, $T = [x^L, x^U] \subset \mathbb{R}^d$, with $x_i^U - x_i^L = D$, for $i = 1, \ldots, d$ and for a given $D > 0$. We discuss how the formulas can be generalised in Subsection 5.1.3.

5.1.1 Bound for ϕ_1 over an Input Box

Consider a given test point x^*, and a generic input point x' in $\mathbb{R}^d$. It is straightforward to notice that $\bar{\boldsymbol{f}}([x^*, x'])$ is still a Gaussian process, as its finite-dimensional distributions are all Gaussian. Consider, then, the matrix $A = [I; -I] \in \mathbb{R}^{m \times 2m}$, where I is the identity matrix of dimension m. For Property 1 we have that the stochastic process

$$\boldsymbol{f}^o(x') := A \cdot \bar{\boldsymbol{f}}([x^*, x']) = \bar{\boldsymbol{f}}(x^*) - \bar{\boldsymbol{f}}(x')$$

is still a Gaussian process. Furthermore, its mean and kernel over a generic input can be computed explicitly and are given by:

$$\begin{aligned} \mu^o(x') &= \bar{\mu}(x^*) - \bar{\mu}(x') \\ \Sigma^o_{x',x''} &= \bar{\Sigma}_{x^*,x^*} + \bar{\Sigma}_{x',x''} - \bar{\Sigma}_{x^*,x''} - \bar{\Sigma}_{x',x^*}, \end{aligned}$$

for x' and $x'' \in \mathbb{R}^d$. Basically, $\boldsymbol{f}^o(x')$ is the stochastic process that describes how the posterior GP changes with respect to its realisation on a test point x^*. For the computation of ϕ_1 we are interested in computing the supremum process of each component of $\boldsymbol{f}^o(x')$, which can then be used to check for the validity of the property h_{conf}, that is, the formula $f_i(x^*) - f_i(x') > \delta$, for a given index $i \in \{1, \ldots, m\}$. In order to do that we apply the Borell-TIS inequality, which gives us an analytical shape for the upper bound random variable, and subsequently employ the Dudley's entropy integral [3, 47] for its computation. To do so, we firstly need to define an

appropriate pseudo-metric space in which to analyse the GP. We denote with $\hat{\boldsymbol{f}}^o(x')$ the zero-centred GP defined as $\hat{\boldsymbol{f}}^o(x') := \boldsymbol{f}^o(x') - \mu^o(x')$. Then, we define:

$$d_i(x', x'') = \sqrt{\mathbb{E}[(\hat{\boldsymbol{f}}_i^o(x') - \hat{\boldsymbol{f}}_i^o(x''))^2]}. \tag{5.1}$$

We assume that the distances induced by the pseudo-metric d_i are bounded by an ℓ_p metric by means of a multiplicative constant $K_{i,p} > 0$, as follows:

$$d_i(x', x'') \leq K_{i,p}||x' - x''||_p \qquad \forall x', x'' \in T. \tag{5.2}$$

We refer to $K_{i,p}$ as the *metric bounding* constant with respect to the metric ℓ_p. Notice that, as all the ℓ_p norms in finite-dimensional vector spaces are strongly equivalent, a value for $K_{i,p}$ found for any $p \in \mathbb{N}$ can be straightforwardly transformed to obtain a valid metric bounding constant, $K_{i,q}$, for any other ℓ_q, with $q \neq p$. Hence, for simplicity of notation we will simply refer to a constant $K_i := K_{i,p}$ in general without referring to the particular metric ℓ_p with respect to which it was computed. We then have that the following theorem holds.

Theorem 1. *Consider a test point x^* and let $T \subseteq \mathbb{R}^d$ be a box with layers of length $D > 0$. Let $i \in \{1, \ldots, m\}$, define:*

$$\mu_i^{o,\sup} := \sup_{x \in T} \mu_i^o(x) \tag{5.3}$$

$$d_i^{\sup} := \sup_{x', x'' \in T} d_i(x', x'') \tag{5.4}$$

$$\xi_i^{\sup} := \sup_{x \in T} (\Sigma_{x,x}^o)_{i,i} \tag{5.5}$$

and consider a metric bounding constant $K_i > 0$ defined as in Equation (5.2). For $\delta > 0$ let

$$\eta_i = \delta - \mu_i^{o,\sup} - 12 \int_0^{\frac{1}{2} d_i^{\sup}} \sqrt{d \ln \left(\frac{\sqrt{d} K_i D}{z} + 1 \right)} dz.$$

Define

$$\hat{\phi}_1^i(x^*, T, \delta) := \begin{cases} e^{-\frac{\eta_i^2}{2\xi_i^{\sup}}} & \text{if} \quad \eta_i > 0 \\ 1 & \text{if} \quad \eta_i \leq 0 \end{cases} \tag{5.6}$$

Then, it holds that

$$\phi_1^i(x^*, T, \delta) \leq \hat{\phi}_1^i(x^*, T, \delta).$$

Proof. If $\eta_i \leq 0$ then $\hat{\phi}_1^i(x^*, T, \delta) = 1$, which is obviously an upper bound for $\phi_1^i(x^*, T, \delta)$, the latter being a probability.

We now consider the case in which $\eta_i > 0$. We proceed by casting the GP as a zero-mean one-dimensional process, to which we can apply the Borell-TIS inequality. In fact, we have that:

$$
\begin{aligned}
&\phi_1^i(x^*, T, \delta) \\
&\qquad \text{(By definition of } \phi_1\text{)} \\
=&P(\exists x \in T \, s.t. \, \big(\bar{\boldsymbol{f}}_i(x) - \bar{\boldsymbol{f}}_i(x^*) > \delta\big) \\
&\qquad \text{(By definition of supremum)} \\
=&P\big(\sup_{x \in T} \boldsymbol{f}_i^o(x) > \delta\big) \\
&\qquad \text{(By linearity of GPs)} \\
=&P\big(\sup_{x \in T} \Big(\hat{\boldsymbol{f}}_i^o(x) + \mu_i^o(x)\Big) > \delta\big) \\
&\qquad \text{(By definition of supremum and of } \mu_i^{o,\text{sup}}\text{)} \\
\leq&P\big(\sup_{x \in T} \hat{\boldsymbol{f}}_i^o(x) > \delta - \mu_i^{o,\text{sup}}\big).
\end{aligned}
$$

We can bound the probability that the supremum random variable $\hat{\boldsymbol{f}}_i^{o,\text{sup}} := \sup_{x \in T} \hat{\boldsymbol{f}}_i^o(x)$ is greater than a given threshold by using the Borell-TIS inequality (see Property 3), so that we have:

$$
P(\hat{\boldsymbol{f}}_i^{o,\text{sup}} > \delta - \mu_i^{o,\text{sup}}) \leq \exp\left(-\frac{\Big(\delta - \mu_i^{o,\text{sup}} - \mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\text{sup}}\right]\Big)^2}{2\xi_i^{\text{sup}}}\right)
$$

as long as the condition:

$$
\delta - \mu_i^{o,\text{sup}} - \mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\text{sup}}\right] > 0 \tag{5.7}
$$

is met. The last bit that we need to compute is then the expected value of the supremum of the GP, $\mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\text{sup}}\right]$, plus verifying that Equation (5.7) holds. This can be achieved by employing Dudley's Entropy Integral [47], and is done in Lemma 1, reported below, in which we obtain the following upper bound by relying on the fact that T is a box:

$$
\mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\text{sup}}\right] \leq 12 \int_0^{\frac{1}{2} d_i^{\text{sup}}} \sqrt{d \ln\left(\frac{\sqrt{d} K_i D}{z} + 1\right)} dz.
$$

By using the definition of η_i and by assumption, we then have that $0 < \eta_i \leq \delta - \mu_i^{o,\sup} - \mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\sup}\right]$, so that the condition in Equation (5.7) is met. Furthermore, we have that:

$$\exp\left(-\frac{\left(\delta - \mu_i^{o,\sup} - \mathbb{E}\left[\hat{\boldsymbol{f}}_i^{o,\sup}\right]\right)^2}{2\xi_i^{\sup}}\right) \leq \exp\left(-\frac{\eta_i^2}{2\xi_i^{\sup}}\right)$$

which proves the theorem statement. □

Lemma 1. *Let $\hat{\boldsymbol{f}}$ be a zero-mean GP defined over the field space $\mathbb{R}^d$, and let $T \subset \mathbb{R}^d$ be an hyper-cube. Consider the generic ith component of the GP with $i \in \{1, \ldots, m\}$, and consider K_i and $d_i^{\sup}$ defined as in the statement of Theorem 1, then:*

$$\mathbb{E}\left[\sup_{x \in T} \hat{\boldsymbol{f}}_i(x)\right] \leq 12 \int_0^{\frac{1}{2}d_i^{\sup}} \sqrt{d \ln\left(\frac{\sqrt{d}K_i D}{z} + 1\right)} dz,$$

where D is the edge-length of T.

Proof. By directly employing the Dudley's entropy integral [3] to the GP $\hat{\boldsymbol{f}}$ we obtain:

$$\mathbb{E}\left[\sup_{x \in T} \hat{\boldsymbol{f}}_i(x)\right] \leq 12 \int_0^{\frac{1}{2}d_i^{\sup}} \sqrt{\ln(N(d_i, z, T))} dz,$$

where $N(d_i, z, T)$ is the smallest number of balls of radius z according to metric d_i (defined in Equation (5.1)) that completely covers T. In particular, in the case in which T is an hyper-cube of edge-length D, we proceed by first computing N for the ℓ_2 metric. That is we proceed by computing $N(\ell_2, r, T)$, which is the number of covering balls of diameter r of T under ℓ_2 norm. As the largest hyper-cube contained inside a $d-$sphere of diameter r has a side of length $\frac{r}{\sqrt{d}}$, we obtain:

$$N(\ell_2, r, T) \leq \left(1 + \frac{D\sqrt{d}}{r}\right)^d.$$

Now, for the definition of K_i we have that:

$$d_i(x', x'') \leq K_i ||x' - x''||_2$$

for all x' and $x'' \in T$. Thus, all the points inside a ball of radius $r = \frac{z}{K_i}$ will have a distance in the d_i metric smaller or equal than z. Hence, the number of covering balls of radius z for T, according to pseudo-metric d_i is upper-bounded by

$$N(d_i, z, T) \leq \left(\frac{\sqrt{d}DK_i}{z} + 1\right)^d$$

which proves the statement of the Lemma. □

In Theorem 1 we derive $\hat{\phi}_1^i(x^*, T, \delta)$ as an upper bound for $\phi_1^i(x^*, T, \delta)$. This provides us with a lower bound (that is, worst-case analysis) for checking probabilistic robustness of the posterior Gaussian process $\bar{\boldsymbol{f}}$. Notice that the bound provides interesting information only as long as $\eta_i > 0$, as it otherwise defaults to 1 (see Equation (5.6)). Intuitively, the requirement $\eta_i > 0$ translates to the requirement that the expected value of the supremum random variable is safe with respect to the property that we are verifying. Of course, in general nothing can be said about the case in which $\eta_i \leq 0$, as we can come up with examples in which the variance of the supremum can be as small as we want it to be, which would make the probability of being robust vanishingly small. In the experiments reported in Section 5.5, we empirically observe $\eta_i > 0$ in many cases that are of interest for applications.

Notice also that in order to be able to explicitly compute $\hat{\phi}_1^i(x^*, T, \delta)$ we need to first compute a set of constants, which are the results of four different optimisation problems, namely: $\mu_i^{o,\sup}$, $d_i^{\sup}$, $\xi_i^{\sup}$ and K_i. In Section 5.2, we present a general framework for the *safe* approximation (i.e., providing formal bounds) of these quantities for GPs in Bayesian inference settings. Before doing this, in the next subsection we show how the upper bound for ϕ_1 can be extended to an upper bound for ϕ_2 by using the union bound of probability.

5.1.2 Bound for ϕ_2 over an Input Box

The overall idea for the computation of ϕ_2 is that of applying the same line of arguments to each of the m output components of the GP as in the case of ϕ_1, and then merging the results together by means of the union bound of probability. This is formalised in the statement below.

Theorem 2. *Consider a test point x^* and let $T \subseteq \mathbb{R}^d$ be a box with layers of length $D > 0$. Define:*

$$\begin{aligned}
\mu^{o,\sup} &:= \sup_{x \in T} ||\mu^o(x)||_p \\
d_i^{\sup} &:= \sup_{x', x'' \in T} d_i(x', x'') \quad \textit{for} \quad i = 1, \ldots, m \\
\xi_i^{\sup} &:= \sup_{x \in T} (\Sigma^o_{x,x})_{i,i} \qquad \textit{for} \quad i = 1, \ldots, m
\end{aligned}$$

and consider metric bounding constants $K_i > 0$, for $i = 1, \ldots, m$, defined as in Equation (5.2). For $\delta > 0$ let

$$\bar{\eta}_i = \frac{\delta - \mu^{o,\sup}}{m} - 12 \int_0^{\frac{1}{2} d_i^{\sup}} \sqrt{d \ln \left(\frac{\sqrt{d} K_i D}{z} + 1 \right)} dz.$$

Define:

$$\hat{\phi}_2(x^*, T, \delta) = \begin{cases} 2\sum_{i=1}^{m} e^{-\frac{\bar{\eta}_i^2}{2\xi_i}} & \text{if} \quad \bar{\eta}_i > 0,\ i = 1, \ldots, m \\ 1 & \text{otherwise} \end{cases}$$

Then, it holds that

$$\phi_2(x^*, T, \delta) \leq \hat{\phi}_2(x^*, T, \delta).$$

Proof. If any of the $\bar{\eta}_i$ is negative then $\hat{\phi}_2$ defaults to 1 and the theorem statement is obviously true.

Consider now the case in which $\bar{\eta}_i > 0$ for all $i \in \{1, \ldots, m\}$. The overall idea is to reduce this case to that of Theorem 1 by using the union bound. We first notice that $\bar{\eta}_i > 0$ for all i implies that $\delta - \mu^{o,\sup} > 0$. Hence:

$$\begin{aligned}
&\phi_2(x^*, T, \delta) = P(\exists x \in T \, s.t. \, ||\bar{\boldsymbol{f}}(x^*) - \bar{\boldsymbol{f}}(x)||_p > \delta) \\
&\qquad \text{(By definition of supremum)} \\
=&P\big(\sup_{x \in T} |\boldsymbol{f}^o(x)| > \delta\big) \\
&\qquad \text{(By definition of } L_p \text{ norm)} \\
=&P\left(\sup_{x \in T} \sqrt[p]{\sum_{i=1}^{m} |\boldsymbol{f}_i^o(x)|^p} > \delta\right) \\
&\qquad \text{(By closure of GPs wrt linear operations and definition of } \mu^{o,\sup} \text{)} \\
\leq&P\left(\sup_{x \in T} \sqrt[p]{\sum_{i=1}^{m} |\hat{\boldsymbol{f}}_i^o(x)|^p} > \delta - \mu^{o,\sup}\right) \\
&\qquad \text{(By the positivity of } |\hat{\boldsymbol{f}}_i^o(x)| \text{ and of } \delta - \mu^{o,\sup}) \\
=&P\left(\sup_{x \in T} \sum_{i=1}^{m} |\hat{\boldsymbol{f}}_i^o(x)|^p > (\delta - \mu^{o,\sup})^p\right) \\
\leq&P\left(\vee_{i \in \{1,\ldots,m\}} \sup_{x \in T} |\hat{\boldsymbol{f}}_i^o(x)|^p > \frac{(\delta - \mu^{o,\sup})^p}{m}\right) \\
&\qquad \text{(By the positivity of } |\hat{\boldsymbol{f}}_i^o(x)| \text{ and of } \delta - \mu^{o,\sup}) \\
=&P\left(\vee_{i \in \{1,\ldots,m\}} \sup_{x \in T} |\hat{\boldsymbol{f}}_i^o(x)| > \frac{(\delta - \mu^{o,\sup})}{\sqrt[p]{m}}\right) \\
&\qquad \text{(By the union bound and symmetric properties of Gaussian distributions)} \\
\leq&2\sum_{i=1}^{n} P\left(\sup_{x \in T} \hat{\boldsymbol{f}}_i^o(x) > \frac{\delta - \mu^{o,\sup}}{\sqrt[p]{m}}\right).
\end{aligned}$$

Exactly as done for Theorem 1, we can apply the Borell-TIS inequality here and Dudley's entropy integral to the last term of the inequality and thus prove the theorem statement. □

As a result of the above theorem, we can derive a formal upper bound $\hat{\phi}_2$ on the probabilistic robustness property associated to ϕ_2. Notice how the role played by each individual $\bar{\eta}_i$ is the same as that played by η_i in the theorem above, and the same rationale then applies to the condition $\bar{\eta}_i > 0$. Notice also that the explicit computation of the bound relies on the same set of constants as for the case of $\hat{\phi}_1$, which here needs to be computed for every i.

5.1.3 Generalisation to Compact Sets

Theorems 1 and 2 have been stated for the case in which T is a box in the input space. However, the theorems can be extended for more general compact sets, at the cost of more complex analytic expression. A way to achieve this is to approach it case by case, computing the explicit solution of the Dudley entropy integral over more complicated input set, that is, by generalising Lemma 1 to classes of sets which are different from boxes. In particular, this requires us to be able to compute (or to over-approximate) $N(\ell_2, r, T)$, i.e. the minimum number of ℓ_2-balls that cover the set T. Lemma 1 tackles the explicit case of input boxes because of the simple analytical formula that results from it, and the fact that those are customarily used for adversarial attacks [77]. However, this can be generalised straightforwardly to generic axis-aligned hyper-rectangles, hyper-spheres, ellipsoids, and similar compact sets T with simple shapes.

Alternatively, for a generic compact set T, the bounds in Theorems 1 and 2 can be used to build safe approximations by using the union bound of probability, as done, e.g., in [207] in the case of probabilistic safety for BNNs. In fact, since T is a compact set, it is also bounded. Hence, it is possible to enclose T in the union of a finite number of interior disjointed hyper-cubes T_l, $l = 1, \ldots, n_L$, that is, such that:

$$T \subseteq \bigcup_{l=1}^{n_L} T_l \qquad \mathring{T}_l \cap \mathring{T}_k = \emptyset \quad l \neq k,$$

and furthermore the over-approximation error can be made vanishingly small. Then,

for a general property h, we have that:

$$\begin{aligned}
&\phi(x^*, T, h) \\
=&P(\exists x \in T \ s.t. \ h(f(x^*), f(x)) > 0) \\
\leq&P(\exists x \in \bigcup_{l=1}^{n_L} T_l \ s.t. \ h(f(x^*), f(x)) > 0) \\
\leq&\sum_{l=1}^{n_L} P(\exists x \in T_l \ s.t. \ h(f(x^*), f(x)) > 0) \\
=&\sum_{i=1}^{n_L} \phi(x^*, T_l, h)
\end{aligned}$$

where the last inequality comes from the application of the union bound.

5.2 Optimisation Framework for GPs

In this section, we build a general framework for the computation of the quantities required for the explicit computation of ϕ_1 and ϕ_2, that is, those defined in Equations (5.2)–(5.5). The method builds on a generalisation of the method employed in [103] for the squared-exponential kernel in the setting of Kriging regression. Specifically, we derive a bounding scheme for the computation of $\mu_i^{o,\sup}$ and $\xi_i^{\sup}$, and use that to compute a fast bound on $d_i^{\sup}$ and K_i. Notice that the required quantities depend only on separate class indexes $i \in \{1, \ldots, m\}$. Hence, for simplicity, in this section we consider a GP with a single output value, $\bar{\boldsymbol{f}} : \Omega \times \mathbb{R}^d \to \mathbb{R}$, and we will omit the explicit dependence on i, with the understanding that this represents just one component of the actual posterior GP that we are interested in bounding. Furthermore, and unless otherwise specified, because the bound in Theorems 1 and 2 relies on T being a box, we will make the slightly more general assumption, in this section, that $T = [x^L, x^U]$ is an axis-aligned hyper-rectangle in the input space $\mathbb{R}^d$. The fact that we explicitly allow for hyper-rectangles will enable us to directly use the scheme obtained here in branch-and-bound settings, where the initial input box T is split into a series of smaller axis-aligned hyper-rectangles.

We start by giving the following definition.

Definition 7 (Bounded Kernel Decomposition)**.** *Consider a one-dimensional kernel function $\Sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a compact set T. We say that (φ, ψ, U) is a bounded decomposition for Σ in T if $\Sigma_{x',x''} = \psi(\varphi(x', x''))$ and the following conditions are satisfied:*

1. *$\varphi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is linearly separable and continuously differentiable along the coordinate lines, so that $\varphi(x', x'') = \sum_{j=1}^{d} \varphi_j(x'_j, x''_j)$;*

2. *$\psi : \mathbb{R} \to \mathbb{R}$ is continuously differentiable and with a finite number of flex points;*

3. *$U : \mathbb{R}^N \times \mathbb{R}^d \times \mathbb{R}^N \to \mathbb{R}$ is an upper bounding function such that for any vector of coefficients $\mathbf{c} = [c_1, \ldots, c_N] \in \mathbb{R}^N$ and finite set of associated input points $[x^{(1)}, \ldots, x^{(N)}] \in \mathbb{R}^N \times \mathbb{R}^d$, with $N \in \mathbb{N}$,[1] we have that $U(\mathbf{c}) \geq \sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)})$.*

Intuitively, a kernel decomposition separates the part of the kernel function that depends on the two inputs (represented by φ) with the part of the kernel that relates their dependence to the variance of the GP (represented by ψ). Assumptions 1 and 2 usually follow immediately from the smoothness of kernel functions used in practice.[2] Assumption 3 guarantees that we are able to upper bound the kernel functions. The key idea is that, thanks to the linearity of the inference equations for GPs, we can then propagate this bound through the inference equations to obtain bounds on the a-posteriori mean and variance of the GP, which are then used to compute bounds on the 4 quantities required for the evaluation of $\hat{\phi}$. Of course, not all the possible kernel functions Σ have kernel decomposition (for example if they are not smooth). However, in Section 5.3 we explicitly compute kernel decompositions (φ, ψ, U) for the main kernel functions used in applications.

Before computing bounds on mean and variance, we prove the following.

Proposition 1. *Let Σ be a kernel and (φ, ψ, U) be a bounded decomposition. Consider a compact set T, then for every $\bar{x} \in \mathbb{R}^d$ there exists a set of real coefficients $\bar{a}_L$, $\bar{b}_L$, $\bar{a}_U$ and $\bar{b}_U$ such that:*

$$g_L(x) := \bar{a}_L + \bar{b}_L \varphi(x, \bar{x}) \leq \Sigma_{x,\bar{x}} \leq \bar{a}_U + \bar{b}_U \varphi(x, \bar{x}) =: g_U(x) \quad \forall x \in T.$$

In other words, g_L and g_U respectively represent an LBF and a UBF on the kernel function, given a fixed input point.

Proof. We show how to compute $\bar{a}_L$ and $\bar{b}_L$; the same arguments can be used for the computation of $\bar{a}_U$ and $\bar{b}_U$ by simply considering $-\Sigma_{x,\bar{x}}$.

[1] In the remainder of this thesis, we will always compute U on the training set points, for simplicity of notation we hence omit the dependence on $[x^{(1)}, \ldots, x^{(N)}]$.

[2] The finite number of flex points can be guaranteed, for example, by inspecting the function derivatives.

Consider $c_L = -1$ and $c_U = 1$ coefficients associated to the input point $\bar{x}$. Let $\varphi_L = U(c_L)$ and $\varphi_U = U(c_U)$, then by Assumption 3 of bounded kernel decomposition we have that $\varphi(x, \bar{x}) \in [\varphi_L, \varphi_U]$ for all $x \in T$. Consider now the function ψ restricted to the interval $[\varphi_L, \varphi_U]$. Then there are four cases to consider for ψ.

Case 1 If ψ happens to be concave in $[\varphi_L, \varphi_U]$, then, by definition of a concave function, a lower bound is given by the line that links the points $(\varphi^L, \psi(\varphi^L))$ and $(\varphi^U, \psi(\varphi^U))$, that is, g_L is simply the LBF with coefficients:

$$\bar{b}_L = \frac{\psi(\varphi^L) - \psi(\varphi^U)}{\varphi^L - \varphi^U}$$
$$\bar{a}_L = \psi(\varphi^L) - \bar{b}_L \varphi^L.$$

Case 2 If ψ happens to be a convex function, then, by definition of convex function and by the differentiability of ψ, a valid lower bound is given by the tangent line in the middle point $\varphi^C = (\varphi^L + \varphi^U)/2$ of the interval, that is, g_L is the LBF with coefficients:

$$\bar{b}_L = \frac{d\psi(\varphi^C)}{d\varphi}$$
$$\bar{a}_L = \psi(\varphi^L) - \bar{b}_L \varphi^L.$$

Case 3 Assume now that ψ is concave in $[\varphi^L, \varphi^F]$, and convex in $[\varphi^F, \varphi^U]$ (the arguments are very similar if we assume the first interval to be the one in which ψ is convex and the second to be the one in which it is concave). In other words, there is only one flex point $\varphi^F \in (\varphi^L, \varphi^U)$. Let $\bar{a}'_L$, $\bar{b}'_L$ be coefficients for linear lower approximation in $[\varphi^L, \varphi^F]$ and $\bar{a}''_L$, $\bar{b}''_L$ analogous coefficients in $[\varphi^F, \varphi^U]$ (respectively computed as for Case 1 and Case 2 above), and call g' and g'' the corresponding functions. Define g_L to be the LBF function of coefficients $\bar{a}_L$ and $\bar{b}_L$ that goes through the two points $(\varphi^L, \min(g'(\varphi^L), g''(\varphi^L)))$ and $(\varphi^U, g''(\varphi^U))$. We then have that g_L is a valid lower bound function for ψ in $[\varphi^L, \varphi^U]$. In order to prove this we distinguish between two cases:

1. if $\min(g'(\varphi^L), g''(\varphi^L)) = g'(\varphi^L)$, then we have that $g_L(\varphi^L) = g'(\varphi^L) \leq g''(\varphi^L)$, and $g_L(\varphi^U) = g''(\varphi^U)$. Hence, because of linearity, $g_L(\varphi) \leq g''(\varphi)$ in $[\varphi^L, \varphi^U]$, and in particular in $[\varphi^F, \varphi^U]$ as well. This also implies that $g_L(\varphi^F) \leq g''(\varphi^F) \leq g'(\varphi^F)$. On the other hand, $g_L(\varphi^L) = g'(\varphi^L)$, hence $g_L(\varphi) \leq g'(\varphi)$ in $[\varphi^L, \varphi^F]$. Combining these two results and by construction of g' and g'' we have that $g_L(\varphi) \leq \psi(\varphi)$ in $[\varphi^L, \varphi^U]$.

2. if $\min(g'(\varphi^L), g''(\varphi^L)) = g''(\varphi^L)$, then in this case we have $g_L = g''$, and just have to show that $g(\varphi) \leq g'(\varphi)$ in $[\varphi^L, \varphi^F]$. This immediately follows by noticing that $g''(\varphi^F) \leq g'(\varphi^F)$ and $g''(\varphi^L) \leq g'(\varphi^L)$.

Case 4 In the general case, as we have a finite number of flex points, we can divide $[\varphi^L, \varphi^U]$ in subintervals in which ψ is either convex or concave. We can then proceed iteratively from the two left-most intervals by repeatedly applying Case 3. □

The above proposition allows us to explicitly compute coefficients of an LBF and a UBF on the overall kernel value, for any fixed point $\bar{x}$ in the input space. The overall idea is that, since the a-posteriori mean and variance is defined in terms of summation and multiplication of pieces of the form $\Sigma_{x,x^{(i)}}$, for all the $x^{(i)}$ in the training dataset $\mathcal{D}$, then we can compute LBFs and UBFs corresponding to each point in the training set, and propagate them through the inference equations for any unseen test point in T. Thanks to the function U we are then able to bound the resulting LBFs and UBFs by the overall mean and variance functions. This is formalised in the following two subsections.

5.2.1 Bounding the A-Posteriori Mean

Concerning the mean function, we want to compute $\mu^{o,\sup} := \sup_{x \in T} \mu^o(x)$, where $\mu^o(x) = \bar{\mu}(x^*) - \bar{\mu}(x)$. As x^* is fixed, we can take $\bar{\mu}(x^*)$ out of the sup calculations so that we have $\mu^{o,\sup} = \bar{\mu}(x^*) - \inf_{x \in T} \bar{\mu}(x)$. In this section we show how to compute a lower bound μ_T^L for the a-posteriori mean function in an axis-aligned hyper-rectangle T, i.e. such that $\mu_T^L \leq \inf_{x \in T} \bar{\mu}(x)$, for a kernel Σ with an associated bounded kernel decomposition (ϕ, ψ, U).[3] In order to do that, we propagate the LBFs and UBFs for the kernels computed as for Proposition 1 through the inference formula for the GP posterior.

For simplicity, we assume that the prior mean function $\mu(x)$ is zero (see Remark 1 for a discussion on the validity of this assumption and how to extend the method presented below to the general case). Then the a-posteriori mean is given by:

$$\bar{\mu}(x) = \Sigma_{x,\mathbf{x}} \mathbf{t} = \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i \tag{5.8}$$

[3] In fact, it is straightforward to see that computing something smaller than the actual infimum provides us with a worst-case scenario and then still produces a valid bound when used for the computation of $\hat{\phi}$.

where $\mathbf{t} = (\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I)^{-1}\mathbf{y}$ is a vector of $\mathbb{R}^N$. Then a lower bound to the mean function can be computed analytically, as stated in the following proposition.

Proposition 2. *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Consider $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$, the set of coefficients for LBFs and UBFs associated to each training point $x^{(i)}$, $i = 1, \ldots, N$ (computed as for Proposition 1). Define:*

$$(\bar{a}_L^{(i)}, \bar{b}_L^{(i)}) = \begin{cases} (a_L^{(i)}, b_L^{(i)}), & \text{if } t_i \geq 0 \\ (a_U^{(i)}, b_U^{(i)}), & \text{otherwise} \end{cases}$$

Then

$$\mu_T^L := \sum_{i=1}^{N} \bar{a}_L^{(i)} + U([\bar{b}_L^{(1)}, \ldots, \bar{b}_L^{(N)}]) \leq \inf_{x \in T} \bar{\mu}(x).$$

Proof. By construction of the $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$ we have that:

$$a_L^{(i)} + b_L^{(i)}\varphi(x, x^{(i)}) \leq \Sigma_{x,x^{(i)}} \leq a_U^{(i)} + b_U^{(i)}\varphi(x, x^{(i)}).$$

By applying Lemma 2, reported below, we then have that:

$$\Sigma_{x,x^{(i)}} t_i \geq \bar{a}_L^{(i)} + \bar{b}_L^{(i)}\varphi(x, x^{(i)}) \quad \forall x \in T. \tag{5.9}$$

Hence, we have that $\sum_{i=1}^{N} \left(\bar{a}_L^{(i)} + \bar{b}_L^{(i)}\varphi(x, x^{(i)})\right)$ is an LBF to the posterior mean. The statement of the theorem then follows directly from the definition of U. □

Lemma 2. *Let $g_L(t) = a_L + b_L t$ and $g^U(t) = g_U(t) = a_U + b_U t$ be an LBF and UBF to a function $g(t)$ $\forall t \in \mathcal{T}$, i.e. $g_L(t) \leq g(t) \leq g_U(t)$ $\forall t \in \mathcal{T}$. Consider two real coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$. Define*

$$\bar{b}_L = \begin{cases} \alpha b_L \ if \alpha \geq 0 \\ \alpha b_U \ if \alpha < 0 \end{cases} \quad \bar{a}_L = \begin{cases} \alpha a_L + \beta \ if \alpha \geq 0 \\ \alpha a^U + \beta \ if \alpha < 0 \end{cases} \tag{5.10}$$

$$\bar{b}_U = \begin{cases} \alpha b_U \ if \alpha \geq 0 \\ \alpha b_L \ if \alpha < 0 \end{cases} \quad \bar{a}_U = \begin{cases} \alpha a_U + \beta \ if \alpha \geq 0 \\ \alpha a_L + \beta \ if \alpha < 0 \end{cases} \tag{5.11}$$

Then:

$$\bar{g}_L(t) := \bar{a}_L + \bar{b}_L t \leq \alpha g(t) + \beta \leq \bar{a}_U + \bar{b}_U t =: \bar{g}_U(t)$$

That is, LBFs can be propagated through linear transformation by redefining the coefficients through Equations (5.10)–(5.11).

The bound on the mean computed in this way scales linearly with the number of training data used, that is, N linear over-approximations are made for the computation of the bound. As such, we expect the bound to be tight when N is small and its quality to decrease when more training samples are used.

Remark 1. *The above proposition was proved in the case in which the prior mean function was assumed to be identically null. In the general case, we notice that the prior mean function has an additive contribution to the posterior mean. In the case of regression, one can proceed by subtracting the prior mean function explicitly from the dataset, so that the GP can be learned by then assuming an identically null prior. Hence, the identically null mean assumption is not restrictive in the case of regression. However, for classification models, the prior is put on the latent space, so that it is not possible to subtract it from the dataset (as the latent space is an artificial construct and it is intrinsic to the learning process itself). In this case, the prior function, if not identically null, needs to be explicitly accounted for in the computation of* μ_T^L. *Thanks to linearity and to Lemma 2, we have that the additive contribution of the prior function is simply added to Equation* (5.9) *from the proof of the proposition above. In general, we need to assume that we are able to compute an LBF to the prior mean function, which can then be combined with* $\bar{a}_L^{(i)} + \bar{b}_L^{(i)}\varphi(x, x^{(i)})$, *so that we are still able to explicitly compute the value for* μ_T^L *using the bounding function* U. *In Chapter 7 we study the specific case in which the prior function for the GP classification model employed there comes from a (non-linear) physiological model, and is hence not-null. Explicit formulas for classes of non-null prior functions used in practice are given in that chapter.*

5.2.2 Variance Computation

For the variance, we want to compute $\xi = \sup_{x \in T} \Sigma^o_{x,x}$. By definition of $\Sigma^o_{x,x}$ and by using the inference equation for the a-posteriori variance we have:

$$\Sigma^o_{x,x} = (\Sigma_{x^*,x^*} + \Sigma_{x,x} - 2\Sigma_{x,x^*}) - (\Sigma_{x^*,\mathbf{x}} S \Sigma^T_{x^*,\mathbf{x}} + \Sigma_{x,\mathbf{x}} S \Sigma^T_{x,\mathbf{x}} - 2\Sigma_{x^*,\mathbf{x}} S \Sigma^T_{x,\mathbf{x}}). \quad (5.12)$$

where $S = (\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I)^{-1}$. The first three terms in the equation above come from the prior distribution, while the second part is the modification due to the dataset observation. In particular, the term $\Sigma_{x,x}$ is the prior variance of the generic input x of T. We define

$$\sigma_p^2 = \max_{x \in T} \Sigma_{x,x} \quad (5.13)$$

i.e., the maximum prior variance in T. For stationary kernels, which are of particular relevance for applications, we have that $\sigma_p^2 = \Sigma_{x,x}$ is equal to a constant value, so that we can simply replace its value in the computation of ξ. In the general case we assume we are able to compute a rough upper-bound such that $\sigma_p^2 \geq \Sigma_{x,x}$ for all $x \in T$. As such, by taking out of the sup computation the terms that do not depend on the optimisation variable x, and by substituting σ_p^2 for $\Sigma_{x,x}$ we then have:

$$\xi \leq \Sigma_{x^*,x^*} - \Sigma_{x^*,\mathbf{x}} S \Sigma_{x^*,\mathbf{x}}^T + \sigma_p^2 - \inf_{x \in T} \left(\Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T + 2\Sigma_{x,x^*} - 2\Sigma_{x^*,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T \right). \quad (5.14)$$

In order to get an upper bound on ξ we only need to obtain a lower bound to the inf in the right-hand-side of Equation (5.14). In the following proposition we show how this can be obtained by solving a convex quadratic programming problem.

Proposition 3. *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Consider $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$, the set of coefficients for LBFs and UBFs associated to each training point $x^{(i)}$, $i = 1, \ldots, N$, and analogously consider a_L^*, b_L^*, a_U^* and b_U^* associated to x^*. Let $\mathbf{r} = [r^{(1)}, \ldots, r^{(N)}]$, r^*, $\varphi^{(i)}$, $\varphi_j^{(i)}$, φ^* and φ_j^*, for $i = 1, \ldots, N$ and $j = 1, \ldots, d$, be slack continuous variables. Let $\bar{\xi}$ be the solution of the following convex quadratic programming problem:*

$$\begin{aligned}
&\inf_{x \in T} \left(\mathbf{r} S \mathbf{r}^T - 2\Sigma_{x^*,\mathbf{x}} S \mathbf{r}^T + 2r^* \right) \\
&\textit{subject to:} \quad r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi^{(i)} \leq 0 \quad i = 1, \ldots, N \\
&\qquad\qquad r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi^{(i)} \leq 0 \quad i = 1, \ldots, N \\
&\qquad\qquad r^* + a_L^* + b_L^* \varphi^* \leq 0 \\
&\qquad\qquad r^* - a_U^* - b_U^* \varphi^* \leq 0 \\
&\qquad\qquad a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} \leq 0 \quad i = 1, \ldots, N \quad j = 1, \ldots, d \\
&\qquad\qquad \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j \leq 0 \quad i = 1, \ldots, N \quad j = 1, \ldots, d \\
&\qquad\qquad a_{j,L}^* + b_{j,L}^* x_j - \varphi_j^* \leq 0 \quad j = 1, \ldots, d \\
&\qquad\qquad \varphi_j^* - a_{j,U}^* - b_{j,U}^* x_j \leq 0 \quad j = 1, \ldots, d \\
&\qquad\qquad \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)} \quad \varphi^* = \sum_{j=1}^{d} \varphi_j^* \quad i = 1, \ldots, N \quad j = 1, \ldots, d
\end{aligned}$$

Then $\xi_T^U := \Sigma_{x^*,x^*} - \Sigma_{x^*,\mathbf{x}} S \Sigma_{x^*,\mathbf{x}}^T + \sigma_p^2 - \bar{\xi}$ is an upper bound for ξ.

Proof. By setting $\mathbf{r} = \Sigma_{x,\mathbf{x}}$ and $r^* = \Sigma_{x,x^*}$ in the infimum computation in Equation (5.14), we obtain the objective function of the problem statement: $\mathbf{r} S \mathbf{r}^T - 2\Sigma_{x^*,\mathbf{x}} S \mathbf{r}^T + 2r^*$, which is quadratic on the variable vector $[\mathbf{r}, r^*]$. Since $\Sigma_{\mathbf{x},\mathbf{x}}$ is a covariance matrix,

it follows that it is positive definite, and hence $S = (\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I)^{-1}$ is a positive definite matrix, which implies that the objective function is a quadratic convex function in the slack variable vector $[\mathbf{r}, r^*]$. In order to obtain a convex program we then need to linearise the constraints $\mathbf{r} = \Sigma_{x,\mathbf{x}}$ and $r^* = \Sigma_{x,x^*}$. We show how this is done for a generic index $i = 1, \ldots, N$; the arguments for r^* are fully analogous.

We have that $r^{(i)} = \Sigma_{x,x^{(i)}} = \psi(\varphi(x, x^{(i)}))$. By Proposition 1 we have that:

$$a_L^{(i)} + b_L^{(i)} \varphi\left(x, x^{(i)}\right) \leq \Sigma_{x,x^{(i)}} \leq a_U^{(i)} + b_U^{(i)} \varphi\left(x, x^{(i)}\right).$$

Hence, the dependence of ψ on the constraints can be linearised by considering the following over-approximation for the definition of $r^{(i)}$:

$$\begin{aligned} r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi\left(x, x^{(i)}\right) &\leq 0 \\ r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi\left(x, x^{(i)}\right) &\leq 0. \end{aligned}$$

The final step is to linearise the dependency over $\varphi\left(x, x^{(i)}\right)$. We introduce slack variables $\varphi^{(i)} = \varphi(x, x^{(i)})$, and $\varphi_j^{(i)} = \varphi_j(x_j, x_j^{(i)})$. For Assumption 1 of Definition 7 we have that $\varphi(x, x^{(i)}) = \sum_{j=1}^{d} \varphi_j(x_j, x_j^{(i)})$. Let $i \in \{1, \ldots, N\}$ and let $j \in \{1, \ldots, d\}$, then by applying Proposition 1 with $\psi := \varphi_j(\cdot, x_j^{(i)})$ and $\varphi := x$, we have that there exists a set of coefficients $a_{j,L}^{(i)}$, $b_{j,L}^{(i)}$, $a_{j,U}^{(i)}$ and $b_{j,U}^{(i)}$ such that:

$$a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j \leq \varphi_j(x_j, x_j^{(i)}) \leq a_{j,U}^{(i)} + b_{j,U}^{(i)} x_j.$$

Hence, we can over-approximate the set of constraints $\varphi^{(i)} = \varphi(x, x^{(i)})$ and $\varphi_j^{(i)} = \varphi(x_j, x_j^{(i)})$ with the following set of linear constraints:

$$\begin{aligned} a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} &\leq 0 \\ \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j &\leq 0 \\ \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)}&. \end{aligned}$$

The formula for ξ_T^U then follows by the definition of infimum and by Equation (5.14). □

Crucially, the proposition above casts the computation of the quantity ξ as the solution of a convex quadratic programming problem, for which ready-made solver software exists [173].

5.2.3 Distance Computation

Concerning the upper bound of the distance function, we want to compute $d^{\sup} := \sup_{x',x''\in T} d(x',x'')$. It would be possible to proceed here in a similar fashion as described in the variance case above and reduce the upper bounding of $d^{\sup}$ to the solution of a convex quadratic optimisation problem (with roughly double the number of variables in Section 5.2.2). However, for computational reasons, we instead rely on the bound on ξ to provide a quick upper bound on $d^{\sup}$. In fact, since d is a semi-distance function, we can employ the triangular inequality and obtain that $d(x',x'') \leq d(x',x^*) + d(x'',x^*)$. For the definition of ξ, and for the symmetry of d, we have that each individual summand is upper-bounded by ξ_T^U, so that we obtain the following.

Proposition 4. *Consider a generic compact set T, and let ξ_T^U be an upper-bound for ξ in T. Then:*

$$d^{\sup} \leq 2\xi_T^U.$$

5.2.4 Metric Bounding

The last quantity we are interested in computing is the metric bounding constant K, which bounds the ratio between the distance d and an arbitrary ℓ_p distance. We derive a safe over-approximation for K that relies solely on the prior distribution of the GP. In fact, we have that:

$$\begin{aligned} d(x',x'') =& \sqrt{\Sigma_{x',x'} + \Sigma_{x'',x''} - 2\Sigma_{x',x''} - \left(\Sigma_{x',\mathbf{x}}^T S \Sigma_{x',\mathbf{x}} + \Sigma_{x'',\mathbf{x}}^T S \Sigma_{x'',\mathbf{x}} - 2\Sigma_{x',\mathbf{x}}^T S \Sigma_{x'',\mathbf{x}}\right)} \\ \leq& \sqrt{\Sigma_{x',x'} + \Sigma_{x'',x''} - 2\Sigma_{x',x''}} \leq \sqrt{2}\sqrt{\sigma_p^2 - \Sigma_{x',x''}} =: \bar{d}(x',x''), \end{aligned} \tag{5.15}$$

with σ_p^2 defined as per Equation (5.13). Because of the above inequality, we have that a bounding constant $\bar{K}$ for the function $\bar{d}$ is still a bounding constant for the pseudo-metric d. In particular, it follows that the statement below holds.

Proposition 5. *Let $\bar{d}$ defined as in Equation* (5.15) *and consider a generic ℓ_p metric $||\cdot||$. Consider:*

$$\bar{K} = \sqrt{\max_{x',x''\in T} 2\frac{\sigma_p^2 - \Sigma_{x',x''}}{||x'-x''||^2}} = \sqrt{\max_{x',x''\in T} 2\frac{\sigma_p^2 - \psi(\varphi(x',x''))}{||x'-x''||^2}}. \tag{5.16}$$

Then $\bar{K}$ is a metric bounding constant for d in T w.r.t. metric $||\cdot||$, that is:

$$d(x',x'') \leq K||x'-x''|| \quad \forall x',x'' \in T.$$

Proposition 5 guarantees that we can then find a valid metric bounding constant by directly solving the optimisation problem of Equation (5.16). This, in general, is not a trivial problem, as it is non-linear and defined over the two d-dimensional continuous variables x' and x''. However, for a stationary kernel over the norm $||\cdot||$ we can select $\varphi(x', x'') = ||x' - x''||$, which reduces the above problem to a single-variable optimisation problem in the slack optimisation variable $\varphi = ||x' - x''||$:

$$\max_{\varphi \in [0, \varphi^U]} \sqrt{2 \frac{\sigma_p^2 - \psi(\varphi)}{\varphi^2}}, \tag{5.17}$$

for suitably computed φ^U in T. The maximum can then be computed by relying on the smoothness of ψ for the computation of the derivatives. Notice that in the non-stationary kernel case the maximisation problem defined in Equation (5.16) is in general unbounded. To see that, it suffices, in fact, to observe that the denominator tends to zero every time that $x' = x''$, while this is generally not the case for the numerator in the non-stationary kernel case. Nevertheless, by performing a normalising transformation of the input space we are still able to restrict Equation (5.16) to the bounded case for problems which are of interest in practice. In particular, the case for the ReLU kernel is discussed in the following section.

Remark 2 (Soundness of our Approach)**.** *Notice that the bounds $\hat{\phi}_1$ and $\hat{\phi}_2$ that we compute in Theorems 1 and 2 are defined for the exact computation of the constants defined in Equations* (5.2)–(5.5)*. While the exact, analytical computation of those is intractable, the approach we developed above provides their safe over-approximations, that is, such that a worst case-scenario is taken into account. As such, the bounds that we actually compute are more pessimistic then what is actually obtained by the exact definitions of $\hat{\phi}_1$ and $\hat{\phi}_2$. With a slight abuse of notation, in the rest of these chapter we will refer to this pessimistic approximation simply as $\hat{\phi}_1$ and $\hat{\phi}_2$.*

5.3 Kernel Function Decomposition

In this section we compute explicit kernel decomposition (φ, ψ, U) for the kernels introduced in Section 3.1.2 and the ReLU deep kernel introduced in Section 3.1.3.

5.3.1 Squared-Exponential Kernel

For the squared-exponential kernel, we build a bounded kernel decomposition by setting:

$$\psi(\varphi) = \sigma^2 \exp(-\varphi)$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2.$$

It is straightforward to notice that Assumptions 1 and 2 are met by this decomposition. Concerning the definition of U, consider a set $x^{(1)}, \ldots, x^{(N)}$ of N points in the input space and associated real coefficients $c_1, \ldots, c_N$. For a hyper-rectangle $T = [x^L, x^U]$ we have that:

$$\begin{aligned}
\sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)}) &= \sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{d} \theta_j (x_j - x_j^{(i)})^2 = \sup_{x \in T} \sum_{j=1}^{d} \theta_j \sum_{i=1}^{N} c_i (x_j - x_j^{(i)})^2 \\
&= \sup_{x \in T} \sum_{j=1}^{d} \left(\theta_j \sum_{i=1}^{N} c_i x_j^2 - 2\theta_j \sum_{i=1}^{N} c_i x^{(i)} x_j + \theta_j \sum_{i=1}^{N} c_i x^{(i)2} \right) \\
&= \sum_{j=1}^{d} \sup_{x_j \in [x_j^L, x_j^U]} \left(\theta_j \sum_{i=1}^{N} c_i x_j^2 - 2\theta_j \sum_{i=1}^{N} c_i x^{(i)} x_j + \theta_j \sum_{i=1}^{N} c_i x^{(i)2} \right).
\end{aligned}$$

The right-hand-side of the last equation simply involves the computation of the maximum of a 1-d parabola over an interval of the real line, which can be done exactly and in constant time by simple inspection of the derivative function and by evaluating the function at the extrema of the interval. Call $\bar{x}_j$ the only critical point of the jth parabola, and denote with $h_j(x_j) = \alpha_j x_j^2 + \beta_j x_j + \gamma_j$ the parabola associated with the jth coordinate value, with $\alpha_j = \theta_j \sum_{i=1}^{N} c_i$, $\beta_j = -2\theta_j \sum_{i=1}^{N} c_i x^{(i)}$ and $\gamma_j = \theta_j \sum_{i=1}^{N} c_i x^{(i)2}$, then we set:

$$U(\mathbf{c}) = \sum_{j=1}^{d} U_j(\mathbf{c}) \tag{5.18}$$

where:

$$U_j(\mathbf{c}) = \begin{cases} \max\{h_j(x_j^L), h_j(x_j^U), h_j(\bar{x}_j)\} & \text{if} \quad \bar{x}_j \in [x_j^L, x_j^U] \\ \max\{h_j(x_j^L), h_j(x_j^U)\} & \text{otherwise} \end{cases}.$$

Finally, as the squared-exponential kernel is stationary, the computation of K follows directly from Equation (5.17).

5.3.2 Rational Quadratic Kernel

An analogous argument to the one above holds for the rational quadratic kernel, where we can set:

$$\psi(\varphi) = \sigma^2 \left(1 + \frac{\varphi}{2}\right)^{-\alpha}$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2.$$

As the definition of φ is exactly the same as for the squared-exponential kernel, then the bounding function U can be defined as in Equation (5.18).

5.3.3 Matérn Kernel

For half-integer values, the explicit form of the Matérn Kernel allows us to find an analogous kernel decomposition to the two discussed above:

$$\psi(\varphi) = \sigma^2 k_p \exp\left(-\sqrt{\hat{k}_p \varphi}\right) \sum_{l=0}^{p} k_{l,p} \sqrt[p-l]{\hat{k}_p \varphi}$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2.$$

5.3.4 Periodic Kernel

For the periodic kernel we define:

$$\psi(\varphi) = \sigma^2 \exp(-0.5\varphi)$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j \sin(p_j (x'_j - x''_j))^2.$$

Assumptions 1 and 2 are trivially satisfied because of the smoothness of ψ and φ. For the definition of the bounding function U we have that:

$$\begin{aligned}\sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)}) &= \sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{d} \theta_j \sin(p_j (x_j - x_j^{(i)}))^2 \\ &\leq \sum_{i=1}^{N} \sum_{j=1}^{d} \sup_{x_j \in [x_j^L, x_j^U]} c_i \theta_j \sin\left(p_j (x_j - x_j^{(i)})\right)^2.\end{aligned}$$

The supremum in the final equation can be obtained by simply inspecting the derivative of $c_i \theta_j \sin\left(p_j (x_j - x_j^{(i)})\right)^2$ and its function value at the extrema of each interval

$[x_j^L, x_j^U]$. Let $U_{ij}(c_i)$ be the value computed in such a way for each i and j, then we define:

$$U(\mathbf{c}) = \sum_{j=1}^{d} \sum_{i=1}^{N} U_{ij}(c_i). \tag{5.19}$$

5.3.5 ReLU Kernel

By noticing that the definition of the ReLU kernel is recursive on the number of layers, for simplicity of notation we limit our discussion to the single layer. The general case can simply be dealt with by iterating the decomposition that we compute below over the number of layers of the deep kernel. For fast computation with the ReLU kernel, we build on the pre-processing procedure outlined by [123].

Formally, we assume that our input variable x is normalised to vary within the coordinate hyper-box $[0,1]^d$, and we map that into the frontier of the unity sphere of $\mathbb{R}^{d+1}$, whose generic point we denote with z, by using hyper-spherical coordinates:

$$\begin{aligned} z_1 &= \cos(\theta_1) \\ z_2 &= \sin(\theta_1)\cos(\theta_2) \\ &\vdots \\ z_{d+1} &= \sin(\theta_1)\ldots\sin(\theta_d) \end{aligned}$$

for $\theta_j \in [0,\pi]$, when $j = 1,\ldots,d-1$, and $\theta_d \in [0,2\pi]$. For simplicity, we then work with hyper-spherical coordinate. We write $\theta = [\theta_1,\ldots,\theta_d]$ for the vector of polar coordinates and $z(\theta)$ for the vector of Cartesian coordinates associated to it, and we indicate the region in which θ can vary with $\Theta = [0,\pi]^{d-1} \times [0,2\pi]$. We define:

$$\begin{aligned} \varphi(z', z'') &= k_1 + k_2 z' \cdot z'' \\ \psi(\varphi) &= \sigma_b^2 + \frac{\sigma_w^2 k_3}{2\pi}\left(\sin\left(\cos^{-1}\varphi\right) + \varphi\left(\pi - \cos^{-1}\varphi\right)\right) \end{aligned}$$

with $k_3 = \sigma_b^2 + \frac{\sigma_w^2}{d+1}$, $k_1 = \frac{\sigma_b^2}{k_3}$ and $k_2 = \frac{\sigma_w^2}{k_3(d+1)}$. By the smoothness of the dot product, the sine and cosine function it follows that Assumptions 1 and 2 of Definition 7 are met by the above decomposition. Concerning the definition of U, let $\theta^{(1)},\ldots,\theta^{(N)}$ be the polar coordinate vectors associated to N points with associated coefficients $c_1,\ldots,c_N$, and consider $T = [\theta^L, \theta^U]$ a hyper-rectangle in Θ, then:

$$\sup_{\theta\in T} \sum_{i=1}^{N} c_i \varphi(\theta, \theta^{(i)}) = k_1 \sum_{i=1}^{N} c_i + k_2 \sup_{\theta\in T}\left[z(\theta) \cdot \left(\sum_{i=1}^{N} c_i z(\theta^{(i)}) \right) \right].$$

Let $\bar{z} = \left(\sum_{i=1}^{N} c_i z(\theta^{(i)})\right)$, then the only thing we are interested in computing is then the supremum of the dot product $z(\theta) \cdot \bar{z} = \sum_{j=1}^{d} z_j(\theta)\bar{z}_j$. To do this, we re-write the dot-product in polar coordinates with the following recursion:

$$\begin{aligned}
z(\theta) \cdot \bar{z} =& g_1(\theta_1, \ldots, \theta_d) \\
g_1(\theta_1, \ldots, \theta_d) =& \bar{z}_1 \cos\theta_1 + \sin\theta_1 g_2(\theta_2, \ldots, \theta_d) \\
& \vdots \\
g_{d-1}(\theta_{d-1}, \theta_d) =& \bar{z}_{d-1} \cos\theta_{d-1} + \sin\theta_{d-1} g_d(\theta_d) \\
g_d(\theta_d) =& \bar{z}_d \cos\theta_d + \bar{z}_{d+1} \sin\theta_d.
\end{aligned}$$

We then proceed iteratively starting from g_d. To find the minimum and maximum of g_d, we notice that it is a one-dimensional real function over a compact interval, $[\theta_d^L, \theta_d^U]$, and has only one critical point corresponding to the zero of its derivative, i.e. $\theta_d^C = \arctan(\bar{z}_{d+1}/\bar{z}_d)$. Then, let M_d and m_d be respectively the maximum and minimum of g_d in $[\theta_d^L, \theta_d^U]$, which can be computed by simply inspecting the values of g_d at the two extrema of the intervals and at the critical point (if this belongs to the interval). We thus define the two auxiliary functions $g_{d-1}^M = \bar{z}_{d-1}\cos\theta_{d-1} + M_d \sin\theta_{d-1}$ and $g_{d-1}^m = \bar{z}_{d-1}\cos\theta_{d-1} + m_d \sin\theta_{d-1}$, whose maximum and minimum, M_{d-1}^M, M_{d-1}^m, m_{d-1}^M, m_{d-1}^m, can be found exactly as for g_d. We then have that $M_{d-1} = \max\{M_{d-1}^M, M_{d-1}^m\}$ and $m_{d-1} = \min\{m_{d-1}^M, m_{d-1}^m\}$ are maximum and minimum values for g_{d-1}. The procedure can then be iterated up until g_1, by propagating maximum and minimum values, M_j and m_j, through each function g_{j-1}. By setting $U(\mathbf{c}) = k_1 \sum_{i=1}^{N} c_i + k_2 M_1$ we then obtain a valid value for the kernel bounding function U.

Notice that the ReLU kernel is not stationary. Still, a valid value for the constant K in this case can be computed by starting from Equation (5.16) as follows. By explicit derivation we have that in the unity sphere of $\mathbb{R}^{d+1}$:

$$\sigma_p^2 = \Sigma_{x,x} = \sigma_b^2 + \frac{\sigma_w^2 k_3}{2}.$$

We also have that on the surface of the hyper-sphere $||x' - x''||^2 = 1 - \alpha_{x',x''}$, where $\alpha_{x',x''} = x' \cdot x''$. Hence, by parameterising the set $T \times T$ with the generic angular coefficients $0 < \alpha < 1$ between a generic $(x', x'') \in T \times T$, $x' \neq x''$, the argument of the maximum in Equation (5.16) boils down to:

$$\frac{\sigma_p^2 - \frac{\sigma_w^2 k_3}{2\pi}\left(\sin\left(\arccos\left(k_1 + k_2\alpha\right)\right) + (k_1 + k_2\alpha)(\pi - \arccos\left(k_1 + k_2\alpha\right))\right)}{1 - \alpha}.$$

We notice that the denominator of the above equation decreases as $\alpha \in (0,1)$ increases, and that the numerator increases as α increases[4]. Hence, by continuity in $(0,1)$, the above equation is upper-bounded by the limit of the function for $\alpha \to 1^-$. By explicit computation we have:

$$\bar{K} = \sqrt{\sigma_w^2 k_3 k_2} = \frac{\sigma_w^2}{\sqrt{d+1}}.$$

5.3.6 Kernel Addition

Consider now the case in which the kernel function Σ is defined by linear composition of two kernels Σ' and Σ'' such as:

$$\Sigma_{x',x''} = k'\Sigma'_{x',x''} + k''\Sigma''_{x',x''} \qquad \forall x', x'' \in \mathbb{R}^d \tag{5.20}$$

for some given k' and $k'' \geq 0$. Then, we have that kernel decomposition for Σ' and Σ'', along with the other quantities that need to be computed, can be simply propagated through the sum. To see that, let (φ', ψ', U') and (φ'', ψ'', U'') be the two kernel decomposition. Then, by simply summing up the LBFs and UBFs for Σ' and Σ'', Proposition 1 can be generalised to this case as follows.

Proposition 6. *Let g'_L, g'_U, g''_L and g''_U be lower and upper bounding function for $\Sigma'_{x,\bar{x}}$ and $\Sigma''_{x,\bar{x}}$, for all $x \in T$, as computed in Proposition 1. Then*

$$g_L(x) = k'g'_L(x) + k''g''_L(x)$$
$$g_U(x) = k'g'_U(x) + k''g''_U(x)$$

are respectively lower and upper bounding functions on $\Sigma_{x,\bar{x}}$.

As a consequence of the above proposition it directly follows that the infimum of the posterior mean function over the compact set T can be safely lower bounded for the kernel Σ by setting:

$$\mu_T^L = k'\mu_T'^L + k''\mu_T''^L,$$

where $\mu_T'^L$ and $\mu_T''^L$ are computed by applying Proposition 2 to the kernels Σ' and Σ''. Similarly, Proposition 3 can be generalised by considering two sets of slack variables, one associated to φ' and one to φ'', and relying directly on the lower and upper bounding functions defined in Proposition 6.

[4]To see this, it suffices to compute the derivative for the ReLU kernel and note that the correlation between two inputs decreases as the angle between the two of them increases.

Concerning the computation of metric bounding constant $\bar{K}$, we notice that $\sigma_p^2 = k'\sigma_p'^2 + k''\sigma_p''^2$ is a valid upper bound on the value of $\sup_{x \in T} \Sigma_{x,x}$, where σ_p' and σ_p'' are the corresponding values for Σ' and Σ''. Then, for every x' and $x'' \in T$ we have that:

$$\begin{aligned}\frac{\sigma_p^2 - \Sigma_{x',x''}}{||x' - x''||} &= \frac{k'\sigma_p'^2 + k''\sigma_p''^2 - k'\Sigma'_{x',x''} - k''\Sigma''_{x',x''}}{||x' - x''||} \\ &= k'\frac{\sigma_p'^2 - \Sigma'_{x',x''}}{||x' - x''||} + k''\frac{\sigma_p''^2 - \Sigma''_{x',x''}}{||x' - x''||} \leq k'\bar{K}' + k''\bar{K}'',\end{aligned}$$

where $\bar{K}'$ and $\bar{K}''$ are metric bounding constants computed for the two building block kernels Σ' and Σ''.

5.3.7 Kernel Multiplication

In the case in which two kernels are combined through multiplication, we have that $\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''}$. This case can be reduced to the addition by considering the two following McCormick's inequalities [138]:

$$\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''} \geq \Sigma'_L\Sigma''_{x',x''} + \Sigma'_{x',x''}\Sigma''_L - \Sigma'_L\Sigma''_L \tag{5.21}$$

$$\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''} \leq \Sigma'_U\Sigma''_{x',x''} + \Sigma'_{x',x''}\Sigma''_L - \Sigma'_U\Sigma''_L, \tag{5.22}$$

where Σ'_L, Σ'_U, Σ''_L and Σ''_U are lower and upper values to Σ' and Σ'' in T, respectively. Then we can proceed by using the kernel summation of Equation (5.21) when computing lower bounding function on the kernel, and Equation (5.22) when computing the upper bounding function, and by using the techniques discussed in the section just above.

5.4 Computational Complexity

After training the GP, the computational cost of computing the bounds on the mean is of the order $\mathcal{O}(m\mathcal{K})$, where m is the output dimension of the GP and $\mathcal{K}$ is the computational cost of computing the kernel bounding function U for each particular kernel considered. For the squared-exponential kernel[5] that scales as $N + d$, where N is the size of the dataset and d is the size of the input space, as can be seen from Equation (5.18). For the periodic kernel have that $\mathcal{K}$ is of the order of Nd, as can be deduced from Equation (5.19). Finally, for the ReLU kernel, by iterating the formulas derived in Section 5.3.5 over the kernel's number of layers, we have that $\mathcal{K}$ scales as $(N + d)L$, where L is the number of layers.

[5] And kernels that have an analogous definition for φ.

The computational cost of the bound on the variance is dominated by the solution of the m convex quadratic optimisation problems defined in Proposition 3 (one for every output dimension). This is polynomial in the problem size, i.e. $2(d+N+Nd+1)$. In practice, some of the problem constraints (and hence some of the slack variables related to them) can be dropped for scalability. In fact, as this enlarges the feasible region of the problems, the computation would still produce a valid bound on ξ. The computation of the metric bounding constant K for the kernels discussed in this chapter, and for the value of d^{sup}, can be done in constant time (for each output dimension), by relying on the analytical expressions that we derived above. Tighter values of K and d^{sup} can be possibly derived, but at the cost of an increased computational time. The applications of the formulas for the kernel sum and kernel multiplication have the effect of doubling each computation. Hence, the computational cost in that case scales with the worst of the two kernel functions Σ' and Σ''.

The computation of the integrals in the statement of Theorem 1 and 2 cannot be done analytically, and hence the computational time of the final bound has an additional cost that depends on the quadrature formula used. In particular, we employ the rectangular integration formula, which is linear in the number of grid points used. Notice that, by relying on the fact that the integrand is a monotonic decreasing function of z, we have that considering the rectangular formula on the first point of each sub-interval of the integration grid will still produce a valid bound on ϕ_1 and ϕ_2.

In general, the tightness of the approximations computed above scales with the size of the input set T, and hence they can all be refined by using a branch-and-bound optimisation technique, which we implement and apply in the experiments discussed in Section 5.5. However, branch and bound has a worst-case cost that is exponential in d, the dimension of the input space.

Interestingly, we notice that the computational time for the computation of probabilistic safety depends strongly on the size of the dataset N. This is an inherent problem of GP Bayesian models; in fact, training and inference with GPs is cubic on the size of the training set. Sparse GP techniques [185] have been developed exactly for this reason; by reducing the effective size of the dataset stored in the GP covariance matrix,[6] they are employed to speed up GP training for large datasets. This also has the effect of additionally speeding up the verification of GP Bayesian models.

[6]That is, either by removing training samples or by building synthetic inducing points.

5.5 Experimental Evaluation

In this section we experimentally investigate the behaviour of the bounds on ϕ_1 and ϕ_2 derived in this chapter in two modelling tasks. We first visualise the quality of the bounds in the 2-dimensional regression task that was introduced in Section 4.1. Then, in Subsection 5.5.2, we rely on weak convergence of wide BNNs to GPs with deep kernel to analyse the probabilistic robustness of deep and wide BNNs in adversarial settings.

5.5.1 2-D Regression Task

We visualise the empirical behaviour of the bounds for ϕ_1 and ϕ_2 on the 2-dimensional regression task, which was introduced in Example 3 of Section 4.1. We employ for the analysis the same settings that we introduced and adopted in Examples 3 and 4 for the statistical estimation of probabilistic adversarial robustness.

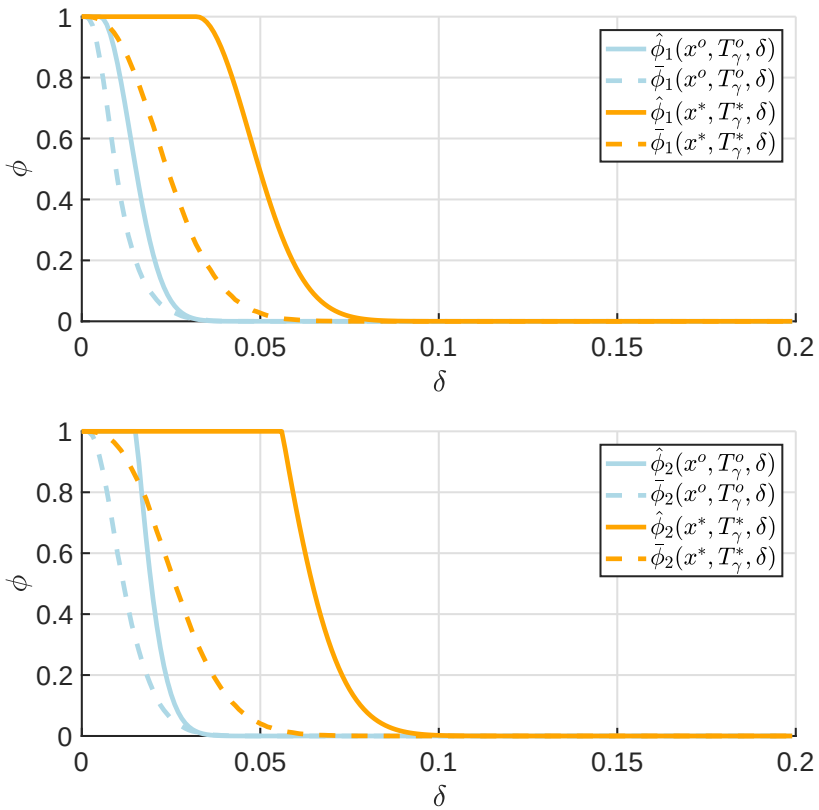


Figure 5.1: Upper bounds (solid lines) and sampling approximation (dashed lines) for ϕ_1 (top plot) and ϕ_2 (bottom plot) on x^o and x^*.

5.5.1.1 Results

Figure 5.1 shows the values obtained for $\hat{\phi}_1$ and $\hat{\phi}_2$ on x^o and x^* for values δ ranging between 0 and 0.2. We observe that values computed for x^* are consistently greater than those computed for x^o, which captures and probabilistically quantifies the increased uncertainty of the GP around x^*, as well as the increased ratio of mean variation around it (see Figure 3.1). Notice also that values for $\hat{\phi}_1$ are always smaller than the corresponding $\hat{\phi}_2$ values. This is a direct consequence of the fact that, for equal values of γ and δ, ϕ_2 is a stronger requirement than ϕ_1 as the latter is not affected by variations that tend to increase the value of the GP output (as that simply translates to increased confidence in classification settings). In Figure 5.1 we also compare the upper bounds obtained with estimation for ϕ_1 and ϕ_2 based on sampling of the GP in a discrete grid around the test points, with the method described in Section 4.1.1. We remark that this provides us with just an empirical under-approximation of the actual values of ϕ_1 and ϕ_2, referred to as $\bar{\phi}_1$ and $\bar{\phi}_2$ respectively. The results suggest that the approximation is tighter around x^o than around x^*. In fact, higher variance will generally imply a looser bound provided by the Borell-TIS inequality, and also due to the over-approximations introduced in the computation of the constants required in the theorems.

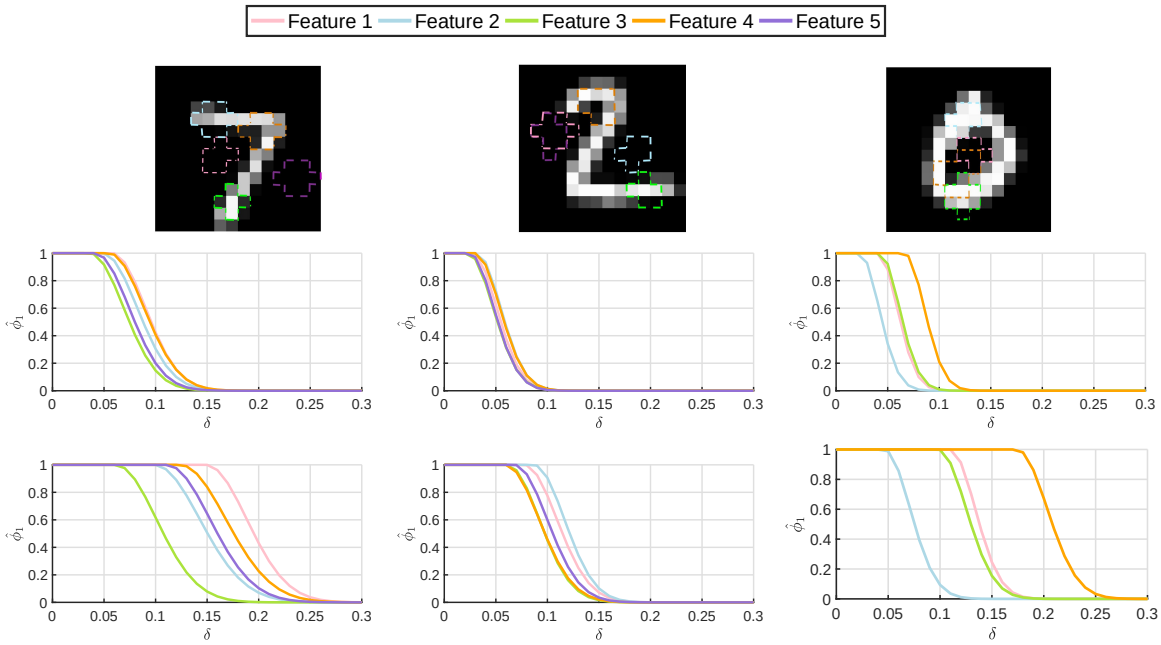


Figure 5.2: First row: three images randomly selected from the MNIST test set, along with detected SIFT features. Second row: respective $\hat{\phi}_1$ values for $\gamma = 0.05$. Third row: respective $\hat{\phi}_1$ values for $\gamma = 0.15$.

5.5.2 BNNs Limit Behaviour on MNIST

We now apply the methods presented above to GPs defined over deep kernels, in an effort to provide a probabilistic analysis of adversarial examples on BNNs. We remark that this analysis is exact for GPs, but only approximate for BNNs, by virtue of weak convergence of the induced distributions between deep kernel GPs and deep BNNs. We focus on GPs with ReLU kernel, which directly correspond to fully-connected BNNs with ReLU activation functions.

Training We follow the experimental setting of [123], that is, we train a selection of ReLU GPs on a subset of the MNIST dataset using least-square classification (i.e. posing a regression problem to solve the classification task) and rely on optimal hyper-parameter values estimated through extensive hyper-parameter search in the latter work. Note that the methods we presented are not constrained to specific kernels or classification models, and can be generalized by suitable modifications to the constant computation part. Classification accuracy obtained on the full MNIST test set varied between 77% (by training only on 100 data samples) to 95% (training on 2000 data samples). Already at 2000 data samples the memory requirements of GP training become prohibitive for standard laptop computers. Larger values of data samples can be considered by using sparse GP methods. This is not explored in this thesis; however, we remark that the verification methods proposed in this chapter adapt without any modifications to the sparse GP settings. Unless otherwise stated, we perform our analyses on the best model obtained using 1000 training samples, that is, a two-hidden-layer architecture with $\sigma_w^2 = 3.19$ and $\sigma_b^2 = 0.00$.

Analysis Settings For scalability purposes we adopt the idea from [206, 175] of performing a feature-level analysis. Namely, we pre-process each image using SIFT [130]. From its output, we keep salient points and their relative magnitude, which we use to extract relevant patches from each image, in the following referred to as *features*. We apply the analysis to thus extracted features. Unless otherwise stated, feature numbering follows the descending order of magnitude (i.e. importance). We first analyse the probabilistic robustness of the GP with the deep kernel when adversarial modifications are applied to the 5 most important features on three images randomly selected from the MNIST test set. We then investigate, in the same setting, the behaviour of the prediction uncertainty on the worst-case perturbations under varying values for the kernel hyper-parameters.

5.5.2.1 Feature-based Robustness Analysis

In the first row of Figure 5.2 we consider three images from the MNIST test data, and for each we highlight the first five features extracted by SIFT (or less if SIFT detected less than five relevant features). For each image $x^{(i)}$, feature f_j and $\gamma > 0$ we consider the set of images $T_{x^{(i)}}^{\mathrm{f}_j,\gamma}$ given by the images differing from $x^{(i)}$ in only the pixels included in f_j and by no more than γ for each pixel.

We plot the values obtained for $\hat{\phi}_1$ as a function of δ for $\gamma = 0.05$ and $\gamma = 0.15$, respectively, on the second and third row of Figure 5.2. Recall that $\hat{\phi}_1$ represents an upper-bound on the probability of finding $x \in T_{x_i}^{\mathrm{f}_j,\gamma}$ such that the classification confidence for the correct class in x drops by more than δ compared to that of $x^{(i)}$. Since a greater γ value implies a larger neighbourhood $T_{x^{(i)}}^{\mathrm{f}_j,\gamma}$, intuitively $\hat{\phi}_1$ will monotonically increase along with the value of γ. Interestingly, the rate of increase is significantly different for different features. In fact, while most of the 14 features analysed in Figure 5.2 have similar $\hat{\phi}_1$ values for $\gamma = 0.05$, the values computed for some of the features using $\gamma = 0.15$ are almost double (e.g. Feature 4 for the third image), and remain fairly similar for others (e.g. Feature 3 for the first image). Also, the relative ordering in robustness for different features is not consistent for different values of γ (e.g. Features 2 and 5 from the first image). This highlights the need of performing parametric analysis of adversarial attacks, which take into account different strengths and misclassification thresholds, as suggested in [16]. Finally, notice that, though only 14 features are explored here, the experiment shows no clear relationship between feature magnitude as estimated by SIFT and feature robustness, which calls for caution in adversarial attacks and defences that rely on black-box feature importance metric. Note also that an empirical analysis of the robustness based on the sampling method discussed in Section 4.1.1, similarly to that performed in Figure 5.1, becomes infeasible for this case study as, in order to have good estimation accuracy, a fine grid over a high-dimensional input space would be required.

5.5.2.2 Variance Analysis

Most active defences are based upon rejecting samples characterised by high uncertainty values. After uncertainty is estimated, defences of this type usually proceed by setting a meta-learning problem whose goal is to distinguish between low and high variance input points, so as to flag potential adversarial examples [85, 56]. However, as we discussed in Chapter 2, mixed results are obtained with this approach [29].

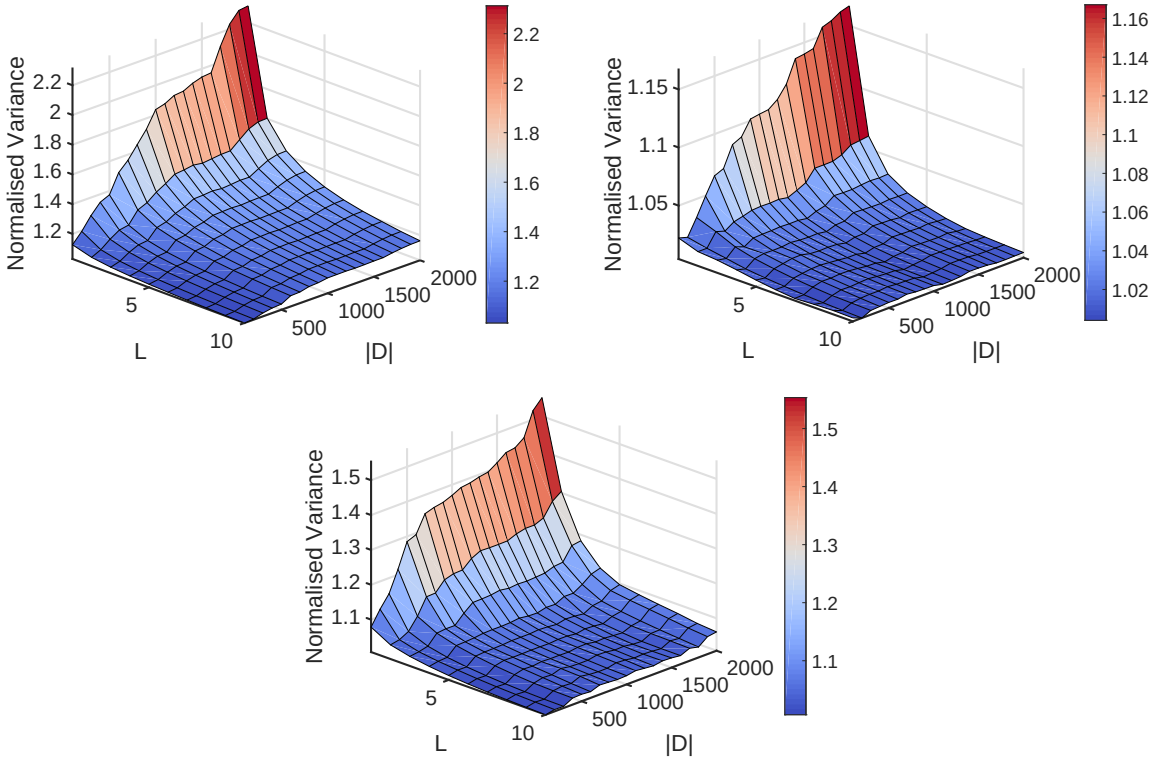


Figure 5.3: Normalized variance $\bar{\sigma}^2$ as a function of L (number of layers of the corresponding BNN) and $|D|$ (number of training points).

In this subsection we aim at analyzing how the variance around test samples changes with different training settings for the three test points discussed previously. We use the method developed for variance optimisation (see Subsection 5.2.2) to compute:

$$\bar{\sigma}^2(x^*) = \frac{1}{\bar{\Sigma}_{x^*,x^*}} \sup_{x \in T_{x^*}^{\mathrm{f}_1,\gamma}} \bar{\Sigma}_{x,x},$$

that is, we look for the highest variance point in the $T_{x^*}^{\mathrm{f}_1,\gamma}$ neighbourhood of x^*, and normalise its value with respect to the variance at x^*. We use $\gamma = 0.15$ and perform the analysis only on Feature 1 (i.e. the most relevant one accordingly to SIFT) of each image.

Figure 5.3 plots values of $\bar{\sigma}^2(x^*)$ as a function of the number of layers of the ReLU kernel (from 1 to 10) and samples (from 100 to 2000) included in the training set. Firstly, notice how maximum values of $\bar{\sigma}^2(x^*)$ are perfectly aligned with the results of Figure 5.2. That is, less robust features are associated with higher values of $\bar{\sigma}^2(x^*)$ (e.g. Feature 1 for image 1). This highlights the relationship between the existence of adversarial examples in the neighbourhood of a point and model uncertainty. We

observe that the normalised variance value consistently monotonically increases with respect to the number of training samples used. This suggests that, as more and more training samples are input into the training model, the latter becomes more confident in predicting "natural" test samples compared to "artificial" ones. Unfortunately, as the number of layers increases, the value of $\bar{\sigma}^2(x^*)$ decreases rapidly to a plateau. This seems to point to the fact that defence methods based on a-posteriori variance thresholding become less effective with more complex neural network architectures, which could be a justification for the mixed results obtained so far using active defences.

5.6 Summary

We presented a formal approach for the probabilistic robustness analysis of Bayesian inference with Gaussian processes with respect to adversarial examples and invariance properties as introduced in Chapter 4. We discussed how the properties considered cannot be computed exactly for a general GP, and have derived a framework for their safe over-approximation.

Specifically, our bounds are based on the Borell-TIS inequality and the Dudley entropy integral, which are theoretically known to give tight bounds for the study of suprema of Gaussian processes [3]. We have then introduced a general framework for the computation (and safe-approximation) of various constants related to the GP posterior. The optimisation framework that we have derived builds on a suitably defined kernel decomposition and the explicit form of GP inference equations so as to derive LBFs and UBFs for the a-posteriori characteristics of the GP. This optimisation framework, used here for the computation of probabilistic safety, will form the backbone of the methodology presented in Chapter 6 for the computation of adversarial robustness of GP Bayesian learning models.

We have next empirically investigated the tightness and the behaviour of our bound on a simple 2-dimensional quadratic regression task, where we observed how the bound varied with different computational parameters and its exponential decay with respect to the size of the input ball. Finally, we have relied upon the weak convergence property of BNNs to GPs with deep kernels and applied the methods developed for the computation of probabilistic robustness of the latter as a way to investigate the robustness of BNNs in adversarial settings. In the MNIST dataset, we have observed how feature importance, as computed by SIFT, is empirically uncorrelated with robustness at the feature level. Interestingly, our optimisation framework

allowed us to perform an analysis of how the worst-case posterior variance is affected by adversarial attacks, which provided us with insights with respect to the behaviour of active defence methodologies used against adversarial examples.

One of the main limitations of our method is that, though providing a safe over-approximation for the GP supremum, the Borell-TIS inequality does not come with a bounding error, so that it is difficult to know in practice how tight the bound is compared to the actual probabilistic safety value for a given GP. Only qualitative statements can, in fact, be made about the tightness of our bound (e.g., it scales with the GP variance and with its - unknown - Lipschitz constants). This compounds in our analysis of BNNs, as the central limit theorem provides convergence only in the limit of infinitely-wide BNNs and only in the sense of weak convergence of distributions. Hence, the observations we made with respect to the behaviour of BNNs on adversarial examples are to be understood in the context of this limit only; and would not hold for, e.g., a network with a bottleneck layer [135]. Interestingly, similar observations about infinitely-wide BNNs have been made in a subsequent work [26], which analysed their adversarial robustness in similar settings. In another follow up work [207], we have developed a method specifically tailored to BNNs for the computation of probabilistic robustness, which just relies on the computation of LBFs and UBFs to the BNN prediction distribution, and can be made arbitrarily close to the actual value by relying on a branch-and-bound scheme. Nevertheless, since this is the first work that analysed probabilistic robustness of a Bayesian learning model in adversarial settings, we believe that our results and methods represent a step towards the application of Bayesian models in safety-critical applications.

Chapter 6

Adversarial Robustness Guarantees for Gaussian Processes

Contents

In this chapter we consider the adversarial robustness of GP models as defined in Section 4.2. Namely, given a compact subset of the input space $T \subseteq \mathbb{R}^d$, representing

a neighbourhood around a test point x^*, a GP trained on a dataset $\mathcal{D}$, and a loss function L we pose the problem of computing guarantees over the optimal decision made by the GP for all points $x \in T$. In particular, we focus on the case in which L is a canonical loss function. In the regression case we thus compute explicit guarantees over the mean of the posterior GP distribution, and show how the methods presented in Chapter 5 for the mean computation can be straightforwardly used in these settings as well.

On the other hand, for classification models we rely on the computation of lower and upper bounds on the posterior predictive distributions. As discussed in Section 4.2.2, in fact, this allows us to give guarantees over changes of classification over input sets T. In order to do so, we proceed by discretising the GP latent space, by means of which we derive an upper and a lower bound on the GP output by analytically optimising a set of Gaussian integrals whose parameters depend upon the extrema of the GP posterior mean and variance in T. Extending the optimisation framework for GPs introduced in Section 5.2, we show how the latter can be bounded by solving a set of convex quadratic and linear programming problems, for which solvers are readily available [23]. Finally, for any given error tolerance $\epsilon > 0$, we prove that there exists a discretisation of the latent space that ensures convergence of the branch and bound to values ϵ-close to the actual maximum and minimum class probabilities in finitely many steps. The method we discuss here is anytime (i.e. the bounds provided are at every step an over-estimation of the actual output ranges over T, and can hence be used to provide guarantees) and ϵ-exact (the actual values are retrieved in finitely many steps up to any arbitrary error $\epsilon > 0$ selected *a-priori*).

We apply our approach to analyse the robustness profile of GP classification models on a two-dimensional dataset, the SPAM dataset, and a feature-based analysis of a binary and a three-class subset of the MNIST dataset. In particular, we compare the guarantees computed by our method with the robustness estimation approximated by adversarial attack methods for GPs; we analyse the effect of approximate Bayesian inference techniques and hyper-parameter optimisation procedures on adversarial robustness; and utilise the methods presented here for interpretability analysis of the GP posterior output. Interestingly, across the three datasets analysed, we observe that approximation based on expectation propagation gives more robust classification models than Laplace approximation, and that GP robustness increases with the number of hyper-parameters training epochs.

This chapter is organised as follows. In the next section, we show how bounds on the classification ranges can be found in the case of two-class classifications. We pro-

vide a branch-and-bound implementation of the method, and prove that the branch-and-bound scheme will converge in finitely many steps to the actual classification ranges, up to any desired error tolerance $\epsilon > 0$. We show in Section 6.2 how the computation for the bound in the multi-class case can be cast as a series of two-class computations by means of iterative conditioning, and use that to derive a safe over-approximation of the classification ranges in the multi-class case. In Section 6.3 we highlight how to deal with GPs learned for regression problems, which requires just a subset of the methods employed for the classification case. The optimisation framework presented in Chapter 5 is extended to the requirements of the method discussed in this chapter in Section 6.4. We show how adversarial robustness can be used for interpretability analysis in Section 6.5. In Section 6.6 we discuss the computational complexity of the method. Experimental results on a 2D classification problem, on the SPAM dataset and on the MNIST dataset are given in Section 6.7.

6.1 Bounding Adversarial Robustness in the Two-Class Classification Case

We start the chapter with a treatment of the two-class classification case. The extension to the multi-class scenario is then described in Section 6.2, and the regression case is discussed in Section 6.3.

As explained in Section 3.2.2, in this case we have a GP over a one-dimensional input space, which greatly simplifies the computations. We assume to be working with a Gaussian analytical approximation technique for the posterior (e.g., Laplace or EP). That is, the posterior GP over the latent space is described by a single output GP of the form: $\bar{\boldsymbol{f}} : \Omega \times \mathbb{R}^d \to \mathbb{R}$, and we denote with $\sigma : \mathbb{R} \to [0, 1]$ its likelihood function. We then omit the output index from the notation that we use in this section.

As we have discussed in Subsection 4.2.2, in order to compute guarantees over the Bayes optimal classifier, we pose the problem of computing the prediction ranges over a generic compact set T, i.e. we aim to compute:

$$\pi_{\min}(T) = \min_{x \in T} \pi(x)$$
$$\pi_{\max}(T) = \max_{x \in T} \pi(x).$$

Similarly to what we have done in Chapter 5, for simplicity of presentation we assume that T is an axis-aligned hyper-rectangle in the input space $\mathbb{R}^d$. The general case can then be obtained by computing over-approximations analogous to those discussed in Section 5.1.3.

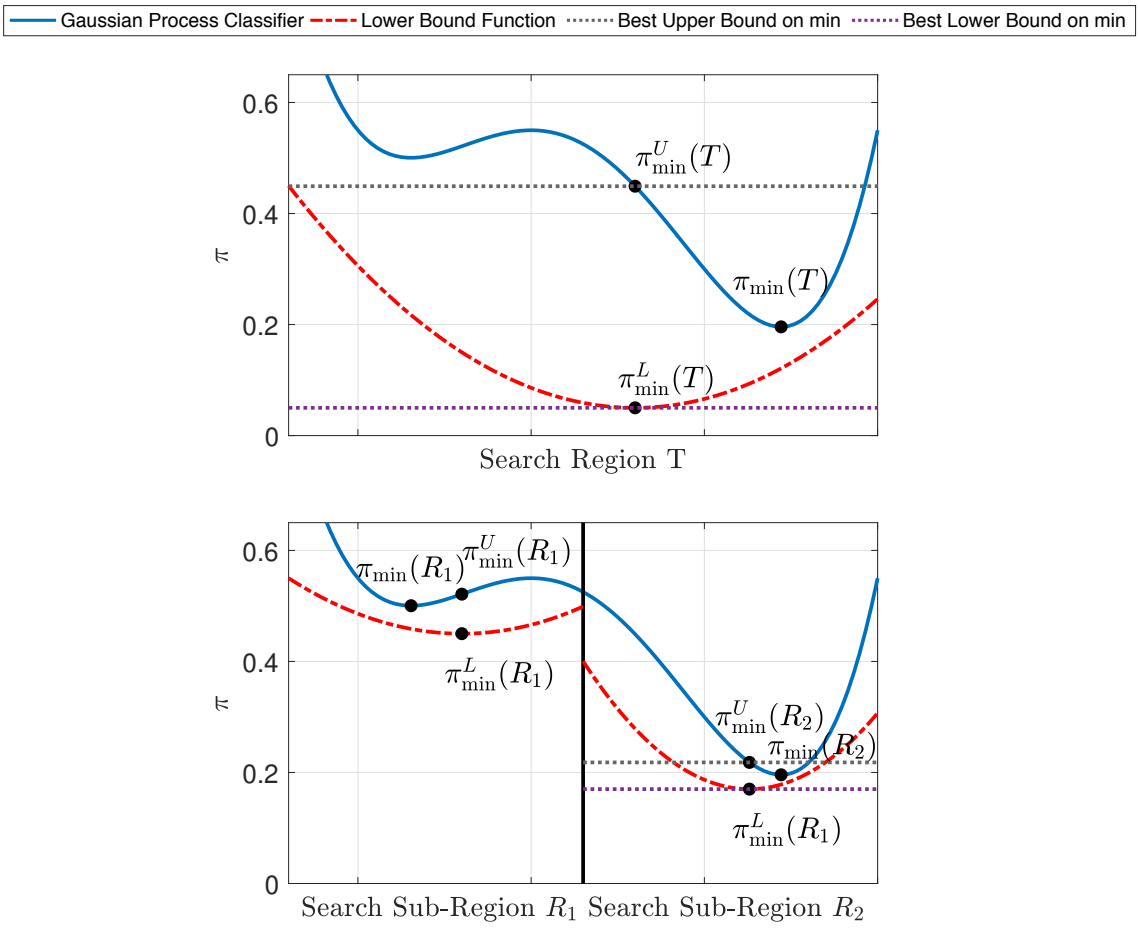


Figure 6.1: **Top:** Computation of upper and lower bounds on $\pi_{\min}(T)$, i.e. the minimum of the classification range on the search region T. **Bottom:** The search region is repeatedly partitioned into sub-regions (only first partitioning visualised), reducing the gap between best lower and upper bounds until convergence (up to ϵ) is reached.

6.1.1 Outline of Approach

An outline of the approach that we use in the two-class classification settings is illustrated in Figure 6.1 for the computation of $\pi_{\min}(T)$ over a one-dimensional set T plotted along the x-axis (the method for the computation of $\pi_{\max}(T)$ is analogous). For any given region T we aim to compute lower and upper bounds for both $\pi_{\min}(T)$ and $\pi_{\max}(T)$, that is, we compute real values $\pi^L_{\min}(T)$, $\pi^U_{\min}(T)$, $\pi^L_{\max}(T)$ and $\pi^U_{\max}(T)$ such that:

$$\pi^L_{\min}(T) \leq \pi_{\min}(T) \leq \pi^U_{\min}(T) \tag{6.1}$$

$$\pi^L_{\max}(T) \leq \pi_{\max}(T) \leq \pi^U_{\max}(T). \tag{6.2}$$

We refer to $\pi^L_{\min}(T)$ and $\pi^U_{\max}(T)$ as *over-approximations* of the ranges, as they provide pessimistic estimation of the actual values of $\pi_{\min}(T)$ and $\pi_{\max}(T)$, and hence tighter guarantees. On the other hand, we refer to $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ as *under-approximations* because they provide an optimistic estimation of the actual values that we want to compute. In order to compute these over- and under-approximations, we compute a lower and an upper bound function (the lower bound function is depicted with a dashed red curve in Figure 6.1) to the GP output (solid blue curve) in the region T. We then find the minimum of the lower bound function, $\pi^L_{\min}(T)$ (shown in the plot), and the maximum of the upper bound function, $\pi^U_{\max}(T)$ (not shown). Then, valid values for $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ can be computed by evaluating the GP predictive distribution on any point in T (a specific $\pi^U_{\min}(T)$ is depicted in Figure 6.1). Finally, we iteratively refine the lower and upper bounds computed in T with a branch-and-bound algorithm. Namely, the region T is recursively subdivided into sub-regions, for which we compute new (tighter) bounds, until these converge up to a desired tolerance $\epsilon > 0$.

6.1.2 Computation of Bounds

In this paragraph we show how to compute $\pi^U_{\max}(T)$, an upper bound on the maximum, and $\pi^L_{\min}(T)$, a lower bound on the minimum of the GP predictive distribution. In order to do so, we work on the assumption that the likelihood function $\sigma(f)$ is a monotonic, non-decreasing, and continuous function of the latent variable (notice that this is trivially satisfied by all the commonly used likelihood functions, e.g., logistic and probit [109]).

By relying on the fact that we are working with Gaussian analytical approximations of the posterior latent distribution, we have that the predictive posterior distribution on a generic point x can be written down as:

$$\pi(x) = \int_{\mathbb{R}} \sigma(\xi)\mathcal{N}(\xi|\bar{\mu}(x), \bar{\Sigma}(x))d\xi \tag{6.3}$$

where $\bar{\mu}$ and $\bar{\Sigma}$ are the posterior mean and variance functions respectively. As discussed in Chapter 3, the above integral cannot be in general solved analytically. Hence, we proceed by performing a discretisation of the latent space. By relying on the linearity of integrals and the monotonicity of the likelihood function σ, we obtain the following proposition, which bounds the extrema of the predictive posterior distribution with a finite sum of weighted Gaussian integrals.

Proposition 7. *Let $\mathcal{S} = \{S_l = [a_l, b_l] \mid l = 1, \ldots, M\}$ be a partition of $\mathbb{R}$ (the latent space) in a finite set of intervals. Then, it holds that:*

$$\pi_{\min}(T) \geq \sum_{l=1}^{M} \sigma(a_l) \min_{x \in T} \int_{a_l}^{b_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \tag{6.4}$$

$$\pi_{\max}(T) \leq \sum_{l=1}^{M} \sigma(b_l) \max_{x \in T} \int_{a_l}^{b_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi. \tag{6.5}$$

Proposition 7 guarantees that the GP predictive distribution in T can be bounded by solving M optimisation problems. Each of these problems seeks to find the a-posteriori mean and variance that maximise or minimise the integral of a Gaussian over an axis-aligned hyper-rectangle T. In order to solve them, we proceed by first assuming the knowledge of lower and upper bounds on the posterior mean and variance in T, that is, μ_T^L, μ_T^U, Σ_T^L and Σ_T^U such that:

$$\mu_T^L \leq \min_{x \in T} \bar{\mu}(x) \quad \mu_T^U \geq \max_{x \in T} \bar{\mu}(x) \tag{6.6}$$

$$\Sigma_T^L \leq \min_{x \in T} \bar{\Sigma}(x) \quad \Sigma_T^U \geq \max_{x \in T} \bar{\Sigma}(x). \tag{6.7}$$

Notice how μ_T^L, μ_T^U, Σ_T^L and Σ_T^U are similar to the quantities that we have computed in Section 5.2 in the case of probabilistic robustness. We will discuss in Section 6.4 how the optimisation framework developed in the context of probabilistic adversarial safety can be adapted for the purposes of this chapter.

By inspection of the derivatives of the integrals in Equations (6.4) and (6.5), the proposition below then follows.

Proposition 8. *Let $\mu^c = \frac{a+b}{2}$ and $\Sigma^c(\mu) = \frac{(\mu-a)^2-(\mu-b)^2}{2 \log \frac{\mu-a}{\mu-b}}$. Then it holds that:*

$$\begin{aligned} &\max_{x \in T} \int_a^b \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \leq \int_a^b \mathcal{N}(\xi | \bar{\mu}^*, \bar{\Sigma}^*) d\xi \\ &= \frac{1}{2} \left(\operatorname{erf} \left(\frac{\bar{\mu}^* - a}{\sqrt{2\bar{\Sigma}^*}} \right) - \operatorname{erf} \left(\frac{\bar{\mu}^* - b}{\sqrt{2\bar{\Sigma}^*}} \right) \right) \end{aligned} \tag{6.8}$$

$$\begin{aligned} &\min_{x \in T} \int_a^b \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \geq \int_a^b \mathcal{N}(\xi | \underline{\mu}^*, \underline{\Sigma}^*) d\xi \\ &= \frac{1}{2} \left(\operatorname{erf} \left(\frac{\underline{\mu}^* - a}{\sqrt{2\underline{\Sigma}^*}} \right) - \operatorname{erf} \left(\frac{\underline{\mu}^* - b}{\sqrt{2\underline{\Sigma}^*}} \right) \right) \end{aligned} \tag{6.9}$$

where we have:

$$\bar{\mu}^* = \arg\min_{\mu \in [\mu_T^L, \mu_T^U]} |\mu^c - \mu|$$

$$\bar{\Sigma}^* = \begin{cases} \Sigma_T^L & \text{if} \quad \bar{\mu}^* \in [a, b] \\ \arg\min_{\Sigma \in [\Sigma_T^L, \Sigma_T^U]} |\Sigma^c(\bar{\mu}^*) - \Sigma| & \text{otherwise} \end{cases}$$

$$\underline{\mu}^* = \arg\max_{\mu \in [\mu_T^L, \mu_T^U]} |\mu^c - \mu|$$

$$\underline{\Sigma}^* = \arg\min_{\Sigma \in \{\Sigma_T^L, \Sigma_T^U\}} [\text{erf}(b|\underline{\mu}^*, \Sigma) - \text{erf}(a|\underline{\mu}^*, \Sigma)]$$

Proof. We provide the proof for the minimum case; similar arguments hold for the maximum.

By definition of μ_T^L, μ_T^U, Σ_T^L, Σ_T^U, we have that for every $x \in T$, $\bar{\mu}(x) \in [\mu_T^L, \mu_T^U]$ and $\bar{\Sigma}(x) \in [\Sigma_T^L, \Sigma_T^U]$. Thus:

$$\min_{x \in T} \int_a^b \mathcal{N}(\xi|\bar{\mu}(x), \bar{\Sigma}(x)) d\xi \geq \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \int_a^b \mathcal{N}(\xi|\mu, \Sigma) d\xi =$$

$$\frac{1}{2} \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \left(\text{erf}\left(\frac{\mu - a}{\sqrt{2\Sigma}} \right) - \text{erf}\left(\frac{\mu - b}{\sqrt{2\Sigma}} \right) \right) = \frac{1}{2} \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \Phi(\mu, \Sigma)$$

where we have set $\Phi(\mu, \Sigma) := \text{erf}\left(\frac{\mu - a}{\sqrt{2\Sigma}}\right) - \text{erf}\left(\frac{\mu - b}{\sqrt{2\Sigma}}\right)$. By looking at the partial derivatives we have that:

$$\frac{\partial \Phi(\mu, \Sigma)}{\partial \mu} = \frac{\sqrt{2}}{\sqrt{\pi \Sigma}} \left(e^{-\frac{(\mu - b)^2}{2\Sigma}} - e^{-\frac{(\mu - a)^2}{2\Sigma}} \right) \geq 0 \Leftrightarrow \mu \leq \frac{a + b}{2} = \mu^c$$

and that if $\mu \notin [a, b]$:

$$\frac{\partial \Phi(\mu, \Sigma)}{\partial \Sigma} = \frac{1}{\sqrt{2\pi\Sigma^3}} \left((\mu - b_i) e^{-\frac{(\mu - b_i)^2}{2\Sigma^2}} - (\mu - a_i) e^{-\frac{(\mu - a_i)^2}{2\Sigma^2}} \right) \geq 0$$

$$\Leftrightarrow \Sigma \leq \frac{(\mu - a)^2 - (\mu - b)^2}{2 \log \frac{\mu - a}{\mu - b}} = \Sigma^c(\mu)$$

otherwise the last inequality has no solutions. As such, μ^c and Σ^c will correspond to global maximum with respect to μ and Σ, respectively. As Φ is symmetric w.r.t. μ^c we have that the minimum value w.r.t. to μ is always obtained for the point furthest away from μ^c, that is, at $\underline{\mu}^* = \arg\max_{\mu \in [\mu_T^L, \mu_T^U]} |\mu^c - \mu|$. The minimum value w.r.t. to Σ will hence be either for Σ_T^L or Σ_T^U, that is $\underline{\Sigma}^* = \arg\min_{\Sigma \in \{\Sigma_T^L, \Sigma_T^U\}} \Phi(\underline{\mu}^*, \Sigma)$. □

In summary, given lower and upper bounds for the a-posteriori mean and variance in T, Proposition 8 allows us to analytically bound the M optimisations of Gaussian

integrals posed by Equations (6.4) and (6.5). Through this, we can compute values for $\pi^L_{\min}(T)$ and $\pi^U_{\max}(T)$, which satisfy the left-hand-side of Equation (6.1) and the right-hand-side of Equation (6.2). Furthermore, note that by definition of $\pi_{\min}(T)$ and $\pi_{\max}(T)$, we have that, for every $x \in T$, setting $\pi^U_{\min}(T) := \pi(x)$ and/or $\pi^L_{\max}(T) := \pi(x)$ provides values which satisfy the right-hand-side of Equation (6.1) and the left-hand-side of Equation (6.2) (we will discuss in Section 6.4 how to pick values of $\bar{x}$ that empirically speed up convergence of branch and bound). Therefore we have the following.

Corollary 1. *Let $\mathcal{S} = \{S_l = [a_l, b_l] \mid l = 1, \ldots, M\}$ be a partition of the latent space $\mathbb{R}$ in a finite set of intervals. Consider μ^L_T, μ^U_T, Σ^L_T, Σ^U_T as defined by Equations* (6.6) *and* (6.7)*, and define for each $l = 1, \ldots, M$ $\bar{\mu}^*_l$, $\bar{\Sigma}^*_l$, $\underline{\mu}^*_l$ and $\underline{\Sigma}^*_l$ to be the solution to the Gaussian integral optimisation problem computed as in Proposition 8. Fix two points $\bar{x} \in T$ and $\underline{x} \in T$, and define:*

$$\pi^L_{\min}(T) := \frac{1}{2}\sum_{l=1}^{M} \sigma(a_l)\left(\operatorname{erf}\left(\frac{\underline{\mu}^*_l - a_l}{\sqrt{2\underline{\Sigma}^*_l}}\right) - \operatorname{erf}\left(\frac{\underline{\mu}^* - b_l}{\sqrt{2\underline{\Sigma}^*_l}}\right)\right)$$

$$\pi^U_{\max}(T) := \frac{1}{2}\sum_{l=1}^{M} \sigma(b_l)\left(\operatorname{erf}\left(\frac{\bar{\mu}^*_l - a_l}{\sqrt{2\bar{\Sigma}^*_l}}\right) - \operatorname{erf}\left(\frac{\bar{\mu}^* - b_l}{\sqrt{2\bar{\Sigma}^*_l}}\right)\right)$$

$$\pi^U_{\min}(T) := \pi(\underline{x})$$

$$\pi^L_{\max}(T) := \pi(\bar{x}).$$

Then $\pi^L_{\min}(T)$, $\pi^U_{\max}(T)$, $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ thus defined satisfy the conditions of Equations (6.1) *and* (6.2)*.*

In Section 6.1.4 we will show how the bounds computed as described in this section can be used to develop a branch-and-bound scheme converging to $\pi_{\min}(T)$ and $\pi_{\max}(T)$. First, however, we notice that discretisation of the latent space performed in this section was necessary because of the fact the predictive posterior distribution could not be written down in explicit form. Interestingly, as discussed in Chapter 3, when the probit function is chosen for the likelihood function, then the integral of Equation (6.3) can be expressed in closed form in terms of the erf function, which leads to a simplification of Proposition 7 and 8. We discuss this particular case in the following section.

6.1.3 Bounds for Probit Classification

For the case that the likelihood σ is taken to be the probit function, that is, $\sigma(\bar{f}) = \Phi(\lambda\bar{f})$ is the cdf of the univariate standard Gaussian distribution scaled by $\lambda > 0$,

we have seen in Chapter 3 that the predictive posterior distribution can be written down as:

$$\pi(x) = \Phi\left(\frac{\bar{\mu}(x)}{\sqrt{\lambda^{-2} + \bar{\Sigma}(x)}}\right).$$

We can use this explicit form to derive analytic upper and lower bounds to the posterior predictive distribution without the need to apply Proposition 7, by relying on upper and lower bounds for the latent mean and variance functions. This can be obtained by direct inspection of the derivatives of $\Phi\left(\frac{\bar{\mu}(x)}{\sqrt{\lambda^{-2}+\bar{\Sigma}(x)}}\right)$. By proceeding similarly to how we did for the proof of Proposition 8 we obtain the following.

Proposition 9. *When the probit likelihood is used for σ, we have that*

$$\pi^L_{\min}(T) := \Phi\left(\frac{\mu^L_T}{\sqrt{\lambda^{-2} + \underline{\Sigma}^*}}\right) \leq \pi_{\min}(T) \tag{6.10}$$

and

$$\pi_{\max}(T) \leq \Phi\left(\frac{\mu^U_T}{\sqrt{\lambda^{-2} + \bar{\Sigma}^*}}\right) =: \pi^U_{\max}(T) \tag{6.11}$$

with

$$\underline{\Sigma}^* = \begin{cases} \Sigma^U_T & \text{if} \quad \mu^L_T \geq 0 \\ \Sigma^L_T & \text{otherwise} \end{cases} \qquad \bar{\Sigma}^* = \begin{cases} \Sigma^L_T & \text{if} \quad \mu^U_T \geq 0 \\ \Sigma^U_T & \text{otherwise.} \end{cases}$$

Crucially the proposition above allows us to avoid performing the discretisation of the latent space in the case of the probit likelihood, which both implies a faster procedure to compute the bound and a tighter bound as well. Notice also that often, even in the case in which the sigmoid likelihood is used, at integration time this is approximated by using a scaled version of the probit with $\lambda = \sqrt{\pi/8}$ (see e.g. [18]). Thus also in these cases we can avoid performing the discretisation of the latent space.

6.1.4 Branch-and-Bound Algorithm

In this paragraph we implement the bounding procedure into a branch-and-bound algorithm and prove convergence up to any a-priori specified $\epsilon > 0$. The overall idea behind branch-and-bound optimisation is that of alternating between bounding the function we are interested in optimising in our search domain T and splitting the search region T into smaller regions, on which we compute the bound in the next iteration. This procedure creates a search tree, in which descending depth implies smaller search regions. The intuition behind the method is that, as we explore the

branch and bound search tree depth-wise, the search regions become smaller, so that the bounds get closer to the true function, and we thus slowly converge to the actual optimum. By computing lower and upper bounds on the quantity of interest, we are then able to prune our search tree for regions on which optimal values cannot occur.

A branch-and-bound scheme built on top of the bounding procedure derived in Section 6.1.2 for the computation of $\pi_{\min}(T)$ is summarised in Algorithm 1, which we now briefly describe (an analogous scheme can be written down for $\pi_{\max}(T)$ as well). After initialising $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ to trivial values and the exploration regions stack $\mathbf{R}$ to the singleton $\{T\}$, the main optimisation loop is entered until convergence (lines 2–9). Among the regions in the stack, we select the region R with the most promising lower bound (line 3), and refine its lower bound using Propositions 7 and 8 (lines 4–5) as well as its upper bound through evaluation of points in R (line 6). If further exploration of R is necessary for convergence (line 7), then the region R is partitioned into two smaller regions R_1 and R_2, which are added to the regions stack and inherit R's bound values (line 8). We do the split by randomly selecting an index $j \in \{1, \ldots, d\}$ among the input dimensions, and by splitting R at the mid-point along the jth dimension. Finally, the freshly computed bounds local to $R \subseteq T$ are used to update the global bounds for T (line 9). Namely, $\pi^L_{\min}(T)$ is updated to the smallest value among the $\pi^L_{\min}(R)$ values for $R \in \mathbf{R}$, while $\pi^U_{\min}(T)$ is set to the lowest observed value yet explicitly computed in line 6.

Algorithm 1 Branch and bound for $\pi_{\min}(T)$

Input: Input space subset T; error tolerance $\epsilon > 0$; latent mean/variance functions $\bar{\mu}(\cdot)$ and $\bar{\Sigma}(\cdot)$.

Output: Lower and upper bounds on $\pi_{\min}(T)$ with $\pi^U_{\min}(T) - \pi^L_{\min}(T) \leq \epsilon$

1: *Initialisation:* Stack of regions $\mathbf{R} \leftarrow \{T\}$; $\pi^L_{\min}(T) \leftarrow -\infty$; $\pi^U_{\min}(T) \leftarrow +\infty$
2: **while** $\pi^U_{\min}(T) - \pi^L_{\min}(T) > \epsilon$ **do**
3: Select region $R \in \mathbf{R}$ with lowest bound $\pi^L_{\min}(R)$ and delete it from stack
4: Find $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R, \Sigma^U_R]$ for latent mean and variance functions over R
5: Compute $\pi^L_{\min}(R)$ from $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R \Sigma^U_R]$ using Corollary 1
6: Find $\pi^U_{\min}(R)$ by evaluating $\pi(x)$ in a point in R
7: **if** $\pi^U_{\min}(R) - \pi^L_{\min}(R) > \epsilon$ **then**
8: Split R into two sub-regions R_1, R_2, add them to stack and use $\pi^L_{\min}(R), \pi^U_{\min}(R)$ as initial bounds for both sub-regions
9: Update $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ with current best bounds found
10: **return** $[\pi^L_{\min}(T), \pi^U_{\min}(T)]$

By construction of the algorithm it is clear that, if it *terminates*, then the resulting values will provide us with an over- and under-approximation of the true value $\pi_{\min}(T)$

with a known error $\epsilon > 0$. Crucially, for our approach to work we have to show that Algorithm 1 converges, i.e. that the loop of lines $2-9$ terminates in a finite number of iterations. In order to prove that that is the case, we rely on the theory of convergence for branch-and-bound algorithms. In particular, to prove convergence of a branch-and-bound scheme up to an error $\epsilon > 0$ it suffices to show that the two following conditions hold [8]:

1. *Consistency Condition*: $\pi^L_{\min}(R) \leq \pi_{\min}(R) \leq \pi^U_{\min}(R) \qquad \forall R \subseteq T.$
2. *Uniform Convergence*: $\forall \epsilon > 0 \quad \exists r > 0 \quad$ s.t. $\forall R \subseteq T$ with $\text{diam}(R) \leq r \Rightarrow |\pi^U_{\min}(R) - \pi^L_{\min}(R)| \leq \epsilon.$

Intuitively, the first condition makes sure that the bounds computed are actually consistent over all the subsets of our initial input region T. It is easy to see that this follows automatically by the construction in Section 6.1.2. The second condition, instead, ensures that the lower and the upper bounds converge uniformly to each other as we reduce the maximum diameter of the branch-and-bound search region to zero. Building on these two conditions we can show the finite-time convergence of our method.

In the following theorem, we show that, for any error $\epsilon > 0$ selected a-priori, there exists a discretisation of the latent function such that the branch-and-bound scheme introduced in Algorithm 1 meets the uniform convergence conditions and hence terminates in a finite number of iterations.

Theorem 3. *Assume that μ^L_R, μ^U_R, Σ^L_R, Σ^U_R are bounding functions for the a-posteriori mean and variance such that:*

$$\mu^L_R \to \min_{x \in R} \bar{\mu}(x), \quad \mu^U_R \to \max_{x \in R} \bar{\mu}(x), \quad \Sigma^L_R \to \min_{x \in R} \bar{\Sigma}(x), \quad \Sigma^U_R \to \max_{x \in R} \bar{\Sigma}(x) \tag{6.12}$$

every time that $\text{diam}(R) \to 0$. Then, for $\epsilon > 0$, there exists a partition of the latent space $\mathcal{S}$ and $\bar{r} > 0$ such that, for every $R \subseteq T$ with $\text{diam}(R) < \bar{r}$, it holds that

$$|\pi^U_{\min}(R) - \pi^L_{\min}(R)| \leq \epsilon. \tag{6.13}$$

Proof. Consider an $\epsilon > 0$, and a generic axis-aligned hyper-rectangle $R \subseteq T$ of diameter $\text{diam}(R) := r > 0$ less than a fixed $\bar{r} > 0$. We want to find a value for $\bar{r}$ for which the condition in Equation (6.13) is surely met. We start by observing that $\pi^U_{\min}(R)$ is defined by computing the predictive posterior distribution on a fixed point of R. Let $\bar{x} \in R$ be such a point, define $\bar{\mu} := \bar{\mu}(\bar{x})$ and $\bar{\Sigma} := \bar{\Sigma}(\bar{x})$, then we have that:

$$\pi^U_{\min}(R) = \int \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}, \bar{\Sigma}) d\xi.$$

Consider now a generic $M > 0$; we define the discretisation of the latent space $\mathcal{S}_M = \{[a_l, b_l] \mid l = 1 \ldots, M\}$ with the following equations:

$$\begin{aligned} a_1 &= -\infty \\ b_l &= \sigma^{-1}\left(\sigma(a_l) + \frac{1}{M}\right) \quad l = 1, \ldots, M \\ a_{l+1} &= b_l \qquad\qquad\qquad\qquad l = 1, \ldots, M, \end{aligned}$$

that is, we discretise the y-axis into M equally distanced intervals and map that discretisation back to the x-axis through the link function, σ^{-1}. We then have that the left-hand-side of Equation (6.13) can be written explicitly as:

$$\left| \int \sigma(\xi)\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi - \sum_{l=1}^{M} \sigma(a_l) \min_{\substack{\mu \in [\mu_R^L, \mu_R^U] \\ \Sigma \in [\Sigma_R^L, \Sigma_R^U]}} \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu, \Sigma)d\xi \right|. \tag{6.14}$$

Let $\mu^{*,(l)}$ and $\Sigma^{*,(l)}$ be the solutions to the lth minimisation problems defined inside the summation of the equation above, then we have:

$$\begin{aligned} &\left| \int \sigma(\xi)\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi - \sum_{l=1}^{M} \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)})d\xi \right| \\ =&\left| \sum_{l=1}^{M} \left(\int_{a_l}^{b_l} \sigma(\xi)\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi - \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)})d\xi \right) \right| \\ \leq&\left| \sum_{l=1}^{M} \left(\left(\sigma(a_l) + \frac{1}{M} \right) \int_{a_l}^{b_l} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi - \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)})d\xi \right) \right| \\ \leq&\left| \frac{1}{M} \sum_{l=1}^{M} \int_{a_l}^{b_l} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi \right| + \left| \sum_{l=1}^{M} \sigma(a_l) \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| \\ \leq&\frac{1}{M} \left| \int_{\mathbb{R}} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma})d\xi \right| + \sum_{l=1}^{M} \sigma(a_l) \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| \\ \leq&\frac{1}{M} + \sum_{l=1}^{M} \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right|. \end{aligned} \tag{6.15}$$

Now, thanks to the conditions in Equation (6.12), we have that as $r \to 0$ both mean and variance converge to the actual maximum and minimum values in R. By further noticing that $\bar{\mu}$ and $\bar{\Sigma}$ are by construction always inside the (vanishing) interval $[\mu_R^L, \mu_R^U] \times [\Sigma_R^L, \Sigma_R^U]$, then for continuity of the Gaussian pdf we have that for each $l = 1, \ldots, M$:

$$\lim_{r \to 0} \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| = 0$$

which means that the second term in Equation (6.15) can be made vanishingly small, in particular less than $\frac{\epsilon}{2}$. By selecting $M = \lceil \frac{2}{\epsilon} \rceil$ the theorem statement holds. $\square$

The above theorem guarantees that, by scaling the number of discretisation points M w.r.t. the inverse of ϵ, the branch-and-bound scheme of Algorithm 1 will converge in a finite number of steps to a solution of the optimisation problem with an error upper-bounded by ϵ. Notice that the larger the M that we select, the smaller the contribution to the bounding error that comes from the discretisation will be. However, a greater value for M also implies that the number of optimisation problems for $\mu^{*,(l)}$ and $\Sigma^{*,(l)}$ to be solved will increase, so a trade-off exists in terms of computational time for the selection of the value of M. Notice also that when the analytical bound for the probit is used, we do not need to perform a discretisation of the latent space. As such, other than just having a faster way of computing the bound, we also obtain that the approximation error incurred in the bounding is smaller, and we would expect faster convergence in this case as well.

We remark that, convergence of the branch-and-bound scheme is linked to the design of converging upper and lower bounds to the a-posterior mean and variance function, that is, meeting the conditions in Equation (6.12). This will be discussed in detail in Section 6.4. First, in the next section, we show how the results obtained here for two-class classification can be extended to the multi-class case.

6.2 Extension to Multiclass Classification

In this section we show how the results for two-class classification can be generalised to the multi-class case. Given a class index $i \in \{1, \ldots, m\}$, we are interested in computing upper and lower bounds on the ith component of the posterior predictive distribution $\pi_i(x)$ for every $x \in T$. In order to do so, we extend Proposition 7 to the multi-class case in Proposition 10, and show that the resulting multi-dimensional integrals can be reduced to the two-class case by conditioning in Proposition 11. Unfortunately, in the multi-class case the use of the multi-dimensional probit likelihood does not lead to any meaningful mathematical simplifications. For simplicity of the arguments, we thus explicitly tackle only the case of the softmax likelihood, but similar arguments can be applied straightforwardly to the case of the multi-dimensional probit, and other likelihood functions that have similar monotonicity properties.

The following is a direct extension of Proposition 7, where the discretisation is performed in a multi-dimensional latent space.

Proposition 10. *Let $\mathcal{S} = \{S_l = [a_l, b_l] \mid l \in \{1, \dots, M\}\}$ be a finite partition of the latent space $\mathbb{R}^m$, with $[a_l, b_l] = [a_{l,1}, b_{l,1}] \times \dots \times [a_{l,m}, b_{l,m}]$. Then, for $i \in \{1, \dots, m\}$:*

$$\pi_{\min,i}(T) \geq \sum_{l=1}^{M} \sigma_i(\underline{\xi}^l) \min_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi$$
$$\pi_{\max,i}(T) \leq \sum_{l=1}^{M} \sigma_i(\bar{\xi}^l) \max_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.$$

where

$$\underline{\xi}^l = [b_{l,1}, \dots, b_{l,i-1}, a_{l,i}, b_{l,i+1}, \dots, b_{l,m}]$$
$$\bar{\xi}^l = [a_{l,1}, \dots, a_{l,i-1}, b_{l,i}, a_{l,i+1}, \dots, a_{l,m}].$$

Proof. We prove the statement for the minimum case; the arguments for the maximum are analogous. By simple properties of integral and definition of minimum we have that:

$$\begin{aligned}\pi_{\min,i}(T) &= \min_{x \in T} \int_{\mathbb{R}^m} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi = \min_{x \in T} \sum_{l=1}^{M} \int_{S_l} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \\ &\leq \sum_{l=1}^{M} \min_{x \in T} \int_{S_l} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.\end{aligned}$$

Taking partial derivatives of the softmax likelihood we have that:

$$\frac{\partial \sigma_i(\xi)}{\partial \xi_k} = \begin{cases} \sigma_i(\xi)(1 - \sigma_i(\xi)) & \text{if } k = i \\ -\sigma_i(\xi)\sigma_k(\xi) & \text{if } k \neq i \end{cases}$$

hence we have that the softmax is monotonically increasing along the direction i and monotonically decreasing along all the other dimensions $k \neq i$. Thus, its minimum in a generic axis-aligned hyper-rectangle $[a_{l,1}, b_{l,1}] \times \dots \times [a_{l,m}, b_{l,m}]$ will occur in the vertex defined as $\underline{\xi}^l = [b_{l,1}, \dots, b_{l,i-1}, a_{l,i}, b_{l,i+1}, \dots, b_{l,m}]$. Thus, we have that the chain of inequalities above can be lower-bounded by computing the softmax on $\underline{\xi}^l$ and taking it outside of the integral computation, which yields:

$$\sum_{l=1}^{M} \sigma_i(\underline{\xi}^l) \min_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.$$

□

In summary, Proposition 10 guarantees that, for all $x \in T$, $\pi_i(x)$ can be upper- and lower-bounded by solving M optimisation problems over a multi-dimensional

Gaussian integral. In Proposition 11, we show that upper and lower bounds for the integral of a multi-dimensional Gaussian distribution, such as those appearing in Proposition 10, can be obtained by optimising a marginalised product of uni-dimensional Gaussian integrals over both the input and the latent space.

We first introduce the following notation. We denote with $\bar{\mu}_{i:j}(x)$ the subvector of $\bar{\mu}(x)$ containing only the components from i to j, with $i \leq j$, and similarly we define $\bar{\Sigma}_{i:k,j:l}(x)$ to be the submatrix of $\bar{\Sigma}(x)$ containing rows from i to k and columns from j to l, with $i \leq k$ and $j \leq l$.

Proposition 11. *Let $S = \prod_{i=1}^{m}[a_i, b_i] \subset \mathbb{R}^m$ be an axis-aligned hyper-rectangle in the latent space. For $i \in \{1, \ldots, m-1\}$ and $f_{\mathcal{I}} \in \mathbb{R}^{m-i-1}$, define $\mathcal{I} = (i+1) : m$ and*

$$\bar{\mu}_i^f(x) = \bar{\mu}_i(x) - \bar{\Sigma}_{i,\mathcal{I}}(x)\bar{\Sigma}_{\mathcal{I},\mathcal{I}}^{-1}(x)(f_{\mathcal{I}} - \bar{\mu}_{\mathcal{I}}(x)) \tag{6.16}$$

$$\bar{\Sigma}_i^f(x) = \bar{\Sigma}_{i,i}(x) - \bar{\Sigma}_{i,\mathcal{I}}(x)\bar{\Sigma}_{\mathcal{I},\mathcal{I}}^{-1}(x)\bar{\Sigma}_{i,\mathcal{I}}^{T}(x). \tag{6.17}$$

Let $S_{\mathcal{I}} = \prod_{j=i+1}^{m}[a_i, b_i]$, then we have that:

$$\max_{x \in T} \int_S \mathcal{N}(\xi|\bar{\mu}(x), \bar{\Sigma}(x))d\xi \leq \max_{x \in T} \int_{a_m}^{b_m} \mathcal{N}(\xi|\bar{\mu}_m(x), \bar{\Sigma}_{m,m}(x))d\xi \prod_{i=1}^{m-1} \max_{\substack{x \in T \\ f \in S_{\mathcal{I}}}} \int_{a_i}^{b_i} \mathcal{N}(\xi|\bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x))d\xi \tag{6.18}$$

$$\min_{x \in T} \int_S \mathcal{N}(\xi|\bar{\mu}(x), \bar{\Sigma}(x))d\xi \geq \min_{x \in T} \int_{a_m}^{b_m} \mathcal{N}(\xi|\bar{\mu}_m(x), \bar{\Sigma}_{m,m}(x))d\xi \prod_{i=1}^{m-1} \min_{\substack{x \in T \\ f \in S_{\mathcal{I}}}} \int_{a_i}^{b_i} \mathcal{N}(\xi|\bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x))d\xi. \tag{6.19}$$

Proof. We consider the minimum case. The maximum follows similarly.

Consider the latent posterior process $\bar{\boldsymbol{f}}$, whose mean and variance function we denote with $\bar{\mu}(x)$ and $\bar{\Sigma}(x)$. Then, we have

$$\min_{x \in T} \int_S \mathcal{N}(\xi|\bar{\mu}(x), \bar{\Sigma}(x))d\xi = \min_{x \in T} P(\bar{\boldsymbol{f}}(x) \in S) = \min_{x \in T} P(a_i \leq \bar{\boldsymbol{f}}_i(x) \leq b_i, i = 1, \ldots, m) =$$

$$\min_{x \in T} P(a_m \leq \bar{\boldsymbol{f}}_m(x) \leq b_m) \prod_{i=1}^{m-1} P(a_i \leq \bar{\boldsymbol{f}}_i(x) \leq b_i | \bar{\boldsymbol{f}}_{\mathcal{I}}(x) \in S_{\mathcal{I}}) \geq$$

(By Lemma 3 reported below)

$$\min_{x \in T} P(a_m \leq \bar{\boldsymbol{f}}_m(x) \leq b_m) \prod_{i=1}^{m-1} \min_{f_{\mathcal{I}} \in S_{\mathcal{I}}} P(a_i \leq \bar{\boldsymbol{f}}_i(x) \leq b_i | \bar{\boldsymbol{f}}_{\mathcal{I}}(x) = f_{\mathcal{I}}) \geq$$

$$\min_{x\in T} P(a_m \leq \bar{\boldsymbol{f}}_m(x) \leq b_m) \prod_{i=1}^{m-1} \min_{\substack{x\in T \\ f_{\mathcal{I}}\in S_{\mathcal{I}}}} P(a_i \leq \bar{\boldsymbol{f}}_i(x) \leq b_i | \bar{\boldsymbol{f}}_{\mathcal{I}}(x) = f_{\mathcal{I}}).$$

Notice that, for each $i \in \{1, \ldots, m-1\}$, $P(a_i \leq \bar{\boldsymbol{f}}_i(x) \leq b_i | \bar{\boldsymbol{f}}_{\mathcal{I}}(x) = f_{\mathcal{I}})$ is the integral of a uni-dimensional Gaussian random variable conditioned on a jointly Gaussian random variable. The statement of the theorem then follows by the application of the conditioning equations for Gaussian distributions stated in Property 2 (Section 3.1.1). □

Lemma 3. *Let X and Y be random variables with joint density function $f_{X,Y}$. Consider measurable sets A and B. Then, it holds that*

$$P(X \in A | Y \in B) \geq \inf_{y\in B} P(X \in A | Y = y).$$

Proof.

$$\begin{aligned} P(X \in A | Y \in B) &= \frac{P(X \in A, Y \in B)}{P(Y \in B)} = \frac{\int_B \int_A f_{X,Y}(x,y)dxdy}{\int_B f_Y(y)dy} \\ &= \frac{\int_B \int_A f_{X|Y}(x|y) f_Y(y)dxdy}{\int_B f_Y(y)dy} \geq \frac{\int_B f_Y(y)dy \inf_{y\in B} \int_A f_{X|Y}(x|y)dx}{\int_B f_Y(y)dy} \\ &= \inf_{y\in B} P(X \in A | Y = y). \end{aligned}$$

□

Proposition 11 reduces the computation of the bounds for the multi-class case to a product of extrema computations over univariate Gaussian distributions, for which Proposition 8 can then be iteratively applied. In order to do that, we need to compute lower and upper bounds to the conditional latent mean and the conditional latent variance defined in Equations (6.16) and (6.17). To do this, we firstly need to compute all the upper and lower bounds to each of the entries of the latent mean vector and latent variance matrix, which we do with the methods described in Section 6.4, and which we denote with $\mu^L_{i,T}$ and $\mu^U_{i,T}$, for $i = 1, \ldots, m$ and $\Sigma^L_{i,j,T}$ and $\Sigma^U_{i,j,T}$, for $i = 1, \ldots, m$ and $j = 1, \ldots, m$. We then notice that Equations (6.16) and (6.17) can be expressed as a rational function in the entries of the mean vector, variance matrix and latent variable vector. After writing them down explicitly we can then propagate the upper and lower bound of each entry down through the rational function equations by simple interval propagation techniques, which results in an upper and lower bound on $\bar{\mu}^f_i(x)$ and $\bar{\Sigma}^f_i(x)$ for $x \in T$ and $f \in S_{\mathcal{I}}$, which we denote with $\mu^{L,f}_{i,T}$, $\mu^{U,f}_{i,T}$, $\Sigma^{L,f}_{i,T}$ and $\Sigma^{U,f}_{i,T}$. This process can then be iterated backward from $i = m$ to $i = 1$, up

until all of the required bounds are computed. Unfortunately, because of the need to symbolically compute a matrix inversion, the explicit formulas for the computation of $\mu_{i,T}^{L,f}$, $\mu_{i,T}^{U,f}$, $\Sigma_{i,T}^{L,f}$ and $\Sigma_{i,T}^{U,f}$ in general are rather convoluted and long (though still consisting of the simple ratio between polynomials). In the following example, we give an explicit treatment of the illustrative case when $m = 3$; the general case can then be obtained by proceeding in a similar fashion for $m > 3$.

Example 8. *Consider the case of three-class classification, that is, with $m = 3$. We have that the integrals associated with $i = m = 3$ in Equations (6.19) and (6.18) are computed with respect to $\bar{\mu}_m(x)$ and $\bar{\Sigma}_{m,m}(x)$, so that for their over-approximation we require only the knowledge of the bounds on the posterior latent mean and variance, $\mu_{m,T}^L$, $\mu_{m,T}^U$, $\Sigma_{m,m,T}^L$ and $\Sigma_{m,m,T}^U$, and no extra computation is required.*

For $i = 2$, we have that $\mathcal{I} = 3$, so that the conditional latent mean and variance are given by:

$$\bar{\mu}_2^f(x) = \bar{\mu}_2(x) - \frac{\bar{\Sigma}_{2,3}(x)}{\bar{\Sigma}_{3,3}(x)}(f_3 - \bar{\mu}_3(x))$$
$$\bar{\Sigma}_2^f(x) = \bar{\Sigma}_{2,2}(x) - \frac{\bar{\Sigma}_{2,3}^2(x)}{\bar{\Sigma}_{3,3}(x)}$$

for $x \in T$ and $f_3 \in S_3 = [a_3, b_3]$. By noticing that by construction we know that $\bar{\mu}_2(x) \in [\mu_{2,T}^L, \mu_{2,T}^L]$, $\bar{\mu}_3(x) \in [\mu_{3,T}^L, \mu_{3,T}^L]$, $\bar{\Sigma}_{2,3}(x) \in [\Sigma_{2,3,T}^L, \Sigma_{2,3,T}^U]$, $\bar{\Sigma}_{3,3}(x) \in [\Sigma_{3,3,T}^L, \Sigma_{3,3,T}^U]$, $\bar{\Sigma}_{2,2}(x) \in [\Sigma_{2,2,T}^L, \Sigma_{2,2,T}^U]$ and $f \in [a_3, b_3]$, we can simply use the formula for interval bound propagation to propagate these 6 input intervals to obtain lower and upper bounds on $\bar{\mu}_2^f$ and $\bar{\Sigma}_2^f$.

For $i = 1$ the equations complicate somewhat, because we now have that $\mathcal{I} = [2, 3]$. Thus, explicit equations for the conditional mean and variance are given by:

$$\bar{\mu}_1^f(x) = \bar{\mu}_1(x) - \frac{\sum_{k=2}^3 \sum_{l=2}^3 \bar{\Sigma}_{1,k}(x)\bar{\Sigma}_{k,l}(x)(f_l - \bar{\mu}_l(x))}{\bar{\Sigma}_{2,2}(x)\bar{\Sigma}_{3,3}(x) - \bar{\Sigma}_{2,3}^2(x)}$$
$$\bar{\Sigma}_2^f(x) = \bar{\Sigma}_{1,1}(x) - \frac{\sum_{k=2}^3 \sum_{l=2}^3 \bar{\Sigma}_{1,k}(x)\bar{\Sigma}_{k,l}(x)\bar{\Sigma}_{1,l}(x)}{\bar{\Sigma}_{2,2}(x)\bar{\Sigma}_{3,3}(x) - \bar{\Sigma}_{2,3}^2(x)}.$$

Notice again that we have upper and lower bounds for all the quantities involved in the computations in the right-hand-side of the equations above. Hence, again by using interval bound arithmetic, we can compute lower and upper bounds for the quantities in the left-hand-side, which is exactly what we need for the bounding of the integrals.

Analogously to what we discussed for the two-class case, the resulting bound can be refined through a branch-and-bound algorithm to ensure convergence up to any desired tolerance $\epsilon > 0$.

Before deriving ways for computing bounds on a-posteriori mean and variance, we first discuss adversarial robustness in the regression case.

6.3 The Case of Regression

While for computing adversarial robustness guarantees for classification models we had to go through the computation of upper and lower bounds on the GP posterior predictive distribution, the analysis is much simpler for the regression case. In fact, using the canonical loss function, we obtain that the optimal decision corresponds with the a-posteriori latent mean function $\bar{\mu}(x)$ of the posterior GP distribution, as discussed in Section 3.2.1. Guarantees over the decision can then be made simply by relying on upper and lower bounds for the mean function, that is, $\mu^L_{i,T}$ and $\mu^U_{i,T}$ for every $i = 1, \ldots, m$. A branch-and-bound scheme can hence be defined directly over the mean function, without the need of bounding the variance or any additional quantities that were needed for the classification case. This makes the computation in the regression case much faster and simpler in practice. In the next section we will discuss how these bounds can be computed, and will prove that as $\text{diam}(T)$ shrinks to zero, they converge uniformly. Thus, a branch-and-bound scheme built on top of these bounds would converge in a finite number of iterations to a solution ϵ-close to the actual one, for any $\epsilon > 0$ selected a-priori.

Additionally, one might be interested in computing similar properties to that computed above over the posterior predictive distribution for the classification case. In the regression case we have noticed that the support of the predictive posterior distribution is always unbounded - as it is a Gaussian distribution - so that reasonable properties to compute include the computation of guarantees that the posterior predictive distribution is above/below a threshold with a given probability. Notice that, as the posterior predictive distribution is itself Gaussian, this problem is exactly the one tackled in Chapter 5, so that these methods discussed there can be directly applied.

6.4 Extension of Optimisation Scheme for GPs

In this chapter we have so far developed ways to give guarantees on the adversarial robustness of a posterior GP, both in case of classification and in case of regression learning models. The methods we have developed relied on the assumption that we were able to compute lower and upper bounds over the extrema of the a-posteriori

mean and variance of the latent GP for any axis-aligned hyper-rectangle of the input space. That is, for each axis-aligned hyper-rectangle $R \subseteq T$, $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, m\}$, we required the knowledge of $\mu^L_{i,R}$, $\mu^U_{i,R}$, $\Sigma^L_{i,j,R}$ and $\Sigma^U_{i,j,R}$, such that:

$$\mu^L_{i,R} \leq \min_{x \in R} \bar{\mu}_i(x) \qquad \mu^U_{i,R} \geq \max_{x \in R} \bar{\mu}_i(x)$$
$$\Sigma^L_{i,j,R} \leq \min_{x \in R} \bar{\Sigma}_{i,j}(x) \qquad \Sigma^U_{i,j,R} \geq \max_{x \in R} \bar{\Sigma}_{i,j}(x).$$

While the development of such bounds is in itself enough for the safe bounding of adversarial robustness, to ensure the convergence guarantees provided by Theorem 3 we also need to show that these bounds are converging, i.e., that the conditions in Equation (6.12) hold true.

In this section we build on the optimisation framework developed for GPs in Chapter 5, and show how it can be adapted for the computations required here, and how convergence can be guaranteed depending on the properties of the kernel decomposition provided. For simplicity we present the framework for GPs built over a single latent output (i.e., capturing the two-class classification case, or single output regression). The extension to the multi-class case is straightforward, it suffices to iterate the computations m times for the mean, and $\frac{m(m+1)}{2}$ for the variance. Notice that, in the variance case when bounding covariances, that is, for $i \neq j$, we will have one kernel decomposition for the kernel of the ith component and one for the kernel of the jth output component. Taking this into account, the rest will follow exactly as in the single output case.

6.4.1 Bounds on A-posteriori Mean

Notice that μ^L_R corresponds exactly to the quantity computed in Proposition 2 of Section 5.2, so that the bounds developed there can here be employed straightforwardly. μ^U_R can be computed in the same way, just by putting a minus sign in front of the a-posteriori mean function $\bar{\mu}$. In this section we prove that the bounds thus provided are converging. First, we prove that the LBFs and UBFs given by Proposition 1 yield converging bounds.

Lemma 4. *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Consider a compact set $T \subset \mathbb{R}^d$, $\bar{x} \in T$, and let, for every axis-aligned hyper-rectangle $R \subseteq T$, $g^R_L(x)$ and $g^R_U(x)$ be the LBF and UBF computed on R for Σ using Proposition 1. Then we have that g^R_L and g^R_U converge uniformly to $\Sigma_{\bar{x},x}$ as $diam(R) \to 0$.*

Proof. We prove the lemma for the LBF. An analogous argument can be made for the UBF.

Let $\epsilon > 0$, we want to find an $\bar{r} > 0$ such that $\text{diam}(R) < \bar{r}$ implies $\max_{x \in R} |g_L^R(x) - \Sigma_{\bar{x},x}| < \epsilon$. Consider φ_L^R and φ_U^R, lower and upper bound values for φ in R. By taking $\bar{r}$ small enough we can assume without loss of generality that $\psi(\varphi)$ has at most one flex point in $[\varphi_L^R, \varphi_U^R]$. We then have the following three cases.

CASE 1: if $\psi(\varphi)$ is concave then g_L^R is defined as the line connecting the two extreme points of the interval $[\varphi_L^R, \varphi_U^R]$. Since $\psi(\varphi)$ is concave, we have that it obtains its minimum in one of these two extrema, so that we have

$$\min_{x \in R} g_L^R(x) = \min_{x \in R} \Sigma_{\bar{x},x}.$$

By Assumption 2 of kernel decompositions (see Definition 7), it follows that ψ is Lipschitz continuous on any compact interval, so that we have that:

$$\lim_{r \to 0} \left| \min_{x \in R} \Sigma_{\bar{x},x} - \max_{x \in R} \Sigma_{\bar{x},x} \right| = 0$$

where $r = \text{diam}(R)$. Putting the two results together we have that the difference between $\min_{x \in R} g_L^R(x)$ and $\max_{x \in R} \Sigma_{\bar{x},x}$ vanishes whenever that r tends to zero, which proves the statement.

CASE 2: if $\psi(\varphi)$ is convex then g_L^R is the Taylor expansion of $\psi(\varphi)$ around the mid-point of the interval, truncated at the first-order. By continuity of φ we then obtain that shrinking r shrinks also the width of the interval $[\varphi_L^R, \varphi_U^R]$, which then, relying on the properties of truncation error of Taylor expansions, proves the lemma statement.

CASE 3: in the case in which a flex point exists, g_L^R is defined to be the maximum line that is below the two LBFs respectively defined over the convex and the concave part of the interval. Since by Case 1 and Case 2 these converge, then we also have that g_L^R converges. □

As the bound that we compute on the mean is obtained by summing together the individual g_L^R computed over each training point $x^{(i)}$, it then follows that convergence of all the g_L^R combined with a tight bounding function U implies convergence of the a-posteriori mean bound. This is formally stated in the proposition below.

Proposition 12. *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Then bounds on the a-posteriori mean μ_R^L and μ_R^U computed through the application of Proposition 2 converge if the bounds provided by U do so.*

Proof. We discuss the case of μ_R^L; the arguments are analogous for μ_R^U.

We have that $\bar{\mu}(x) = \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i$. By Proposition 2, we have that:

$$\sum_{i=1}^{N} t_i \bar{g}_L^{(i)}(x) \leq \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i \tag{6.20}$$

where $\bar{g}_L^{(i)}(x) = g_L^{(i)}(x)$ if $t_i \geq 0$ and $\bar{g}_L^{(i)}(x) = g_U^{(i)}(x)$ otherwise. For Lemma 4 we have that each $g_L^{(i)}$ converges uniformly to $\Sigma_{x,x^{(i)}}$ for each $x^{(i)}$. As t_i is a scalar quantity then we also have that each $t_i \bar{g}_L^{(i)}(x)$ converges uniformly to $\Sigma_{x,x^{(i)}} t_i$. Hence, we have that the bounds in Equation (6.20) converge uniformly as $\text{diam}(R) = r \to 0$, by virtue of being a linear combination of bounds that converge uniformly. The statement of the proposition then follows by the definition of U. □

Hence, convergence of the bounds on the a-posteriori mean and variance is reduced to a property of the kernel bounding function U.

Remark 3. *In Section 5.3 we have computed explicit kernel decomposition for many kernel functions used in practice. It is easy to see that the functions U provided in that section converge to the actual values requested. In fact, the computation of U is actually exact in all the cases discussed there except for the periodic kernel, where the over-approximation comes from swapping the minimum and the sum computations. In this case, convergence of the bound provided by U then follows easily from the linearity of summations. Similarly, for the addition and multiplication formulas of the kernels, we have that convergence of U follows from the convergence of each individual sub-kernel bounding function.*

6.4.2 Bounds on A-posteriori Variance

In Section 5.2.2 we were interested in computing an upper-bound on the variance of the difference of the GP outputs between a given test point and a generic input point in T. For the computation of adversarial robustness we are instead interested in just bounding the variance of a generic input point itself. This leads us to a simpler form of the function that we are optimising and leaves us with only two terms from Equation (5.12), that is $\Sigma_{x,x} - \Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T$, where $S = \Sigma_{\mathbf{x},\mathbf{x}}^{-1}$. For simplicity we assume that $\Sigma_{x,x} = \sigma_p^2$ for all $x \in T$,[1] so that we are only interested in computing:

$$\sigma_p^2 - \min_{x \in T} \Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T \qquad\qquad \sigma_p^2 + \min_{x \in T} -\Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T.$$

[1] If this is is not the case, then $\Sigma_{x,x}$ can be replaced by either its maximum or minimum value according to whether we want to compute the minimum or the maximum of the a-posteriori variance, similarly to what we did in Section 5.2.2.

Upper Bound of Variance The computation of the maximum (that is, the left-hand-side of the equation above) follows exactly like the computations performed in Section 5.2.2. In fact, here we just have a simpler form for the objective function and a reduced number of constraints. Therefore, by proceeding similarly to what we did for Proposition 3 one can prove the following.

Proposition 13. *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Consider $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$, a set of coefficients for LBFs and UBFs associated to each training point $x^{(i)}$, $i = 1, \ldots, N$. Let $\mathbf{r} = [r^{(1)}, \ldots, r^{(N)}]$, $\varphi^{(i)}$, $\varphi_j^{(i)}$, for $i = 1, \ldots, N$ and $j = 1, \ldots, d$, be slack continuous variables. Let $\bar{\sigma}^2$ be the solution of the following convex quadratic programming problem:*

$$
\begin{aligned}
&\min_{x \in T} \mathbf{r} S \mathbf{r}^T \\
&\textit{subject to:} \quad r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi^{(i)} \leq 0 \quad i = 1, \ldots, N \\
&\qquad\qquad r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi^{(i)} \leq 0 \quad i = 1, \ldots, N \\
&\qquad\qquad a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} \leq 0 \quad i = 1, \ldots, N \quad j = 1, \ldots, d \\
&\qquad\qquad \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j \leq 0 \quad i = 1, \ldots, N \quad j = 1, \ldots, d \\
&\qquad\qquad \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)} \qquad\qquad i = 1, \ldots, N \quad j = 1, \ldots, d.
\end{aligned}
$$

Then $\Sigma_T^U := \sigma_p^2 - \bar{\sigma}^2$ is an upper bound for the posterior variance $\bar{\Sigma}_{x,x}$ in T.

Lower Bound of Variance The situation is unfortunately more complicated for the computation of $\min_{x \in T} -\Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T$. In fact, though we can write down an optimisation problem akin to that of Proposition 13, since S is positive definite we have that $-S$ is negative definite, which means that the function we would want to optimise is quadratic concave in that case. Thus, a number of local minima may exist, and simple quadratic optimisation would not be guaranteed to obtain the global solution. However, as we are interested in dealing with worst-case scenario analyses, we need to compute the global minimum. This is unfortunately an NP-hard problem, whose exact solution would make a branch-and-bound algorithm based on it impractical.

Instead, we follow the methods discussed in [173] and proceed by computing a safe lower bound to that, that is, we want to compute a lower bound to the solution

of:

$$
\begin{aligned}
&\min_{x\in T} -\mathbf{r}S\mathbf{r}^T \\
&\text{subject to:} \quad r^{(i)} + a_L^{(i)} + b_L^{(i)}\varphi^{(i)} \leq 0 \quad i = 1,\dots,N \\
&\qquad\qquad r^{(i)} - a_U^{(i)} - b_U^{(i)}\varphi^{(i)} \leq 0 \quad i = 1,\dots,N \\
&\qquad\qquad a_{j,L}^{(i)} + b_{j,L}^{(i)}x_j - \varphi_j^{(i)} \leq 0 \quad i = 1,\dots,N \quad j = 1,\dots,d \\
&\qquad\qquad \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)}x_j \leq 0 \quad i = 1,\dots,N \quad j = 1,\dots,d \\
&\qquad\qquad \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)} \qquad\qquad i = 1,\dots,N \quad j = 1,\dots,d
\end{aligned}
$$

We highlight the details of the procedure applied to our specific setting below. First, we start by re-writing the constraints of the optimisation problem above in matrix form. We introduce the aggregate variable vector $\mathbf{z} = [x_1,\dots,x_d,\varphi^{(1)},\dots,\varphi^{(N)},\varphi_1^{(1)},\dots,\varphi_d^{(N)}]$. Since the constraints are linear, it is possible to define two matrices A_r and A_z such that the optimisation problem above can be equivalently written down as:

$$
\begin{aligned}
&\min -\mathbf{r}^T S\mathbf{r} \\
&\text{Subject to:} \quad A_r\mathbf{r} + A_z\mathbf{z} \leq b \\
&\qquad\qquad \mathbf{r}^L \leq \mathbf{r} \leq \mathbf{r}^U \\
&\qquad\qquad \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U
\end{aligned}
\tag{6.21}
$$

for suitably defined vectors b, $\mathbf{r}^L$, $\mathbf{r}^U$, $\mathbf{z}^L$, $\mathbf{z}^U$. Now, as S is symmetric and positive definite, there exists a matrix of eigenvectors $U = [\mathbf{u}^{(1)},\dots,\mathbf{u}^{(N)}]$ and a diagonal matrix of the associated eigenvalues $\lambda^{(i)}$, for $i = 1,\dots,N$, Λ, such that $S = U\Lambda U^T$. We hence define $\hat{r}^{(i)} = \mathbf{u}^{(i)}\cdot\mathbf{r}$ for $i = 1,\dots,N$, the rotated variables, and $\hat{\mathbf{r}}$ the aggregated vector of rotated variables, and compute their ranges $[\hat{r}^{(i),L},\hat{r}^{(i),U}]$ by solution of the following $2N$ linear programming problems:

$$
\begin{aligned}
&\min / \max \quad \mathbf{u}^{(i)}\cdot\mathbf{r} \\
&\text{Subject to:} \quad A_r\mathbf{r} + A_z\mathbf{z} \leq b \\
&\qquad\qquad \mathbf{r}^L \leq \mathbf{r} \leq \mathbf{r}^U \\
&\qquad\qquad \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U.
\end{aligned}
$$

Implementing the change of variables into the optimisation problem defined in Equa-

tion (6.21) we obtain:

$$\begin{aligned} \min & -\hat{\mathbf{r}}^T \Lambda \hat{\mathbf{r}} \\ \text{Subject to:} \quad & A_{\hat{r}} \hat{\mathbf{r}} + A_z \mathbf{z} \leq b \\ & \hat{\mathbf{r}}^L \leq \hat{\mathbf{r}} \leq \hat{\mathbf{r}}^U \\ & \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U \end{aligned}$$

where we have set $A_{\hat{r}} = A_r U$. We then notice that $\hat{\mathbf{r}}^T \Lambda \hat{\mathbf{r}} = \sum_{i=1}^{N} \lambda^{(i)} \hat{r}^{(i)2}$. Each summand is a simple one-dimensional quadratic function, for which we can find a linear LBF by relying on Case 1 or Case 2 from Proposition 1. Let $\alpha^{(i)}$ and $\beta^{(i)}$ be coefficients of such LBFs, then we have that $\alpha^{(i)} + \beta^{(i)} \hat{r}^{(i)} \leq -\lambda^{(i)} \hat{r}^{(i),2}$ for all $i = 1, \ldots, N$. Let $\boldsymbol{\beta} = [\beta^{(1)}, \ldots, \beta^{(N)}]$ and $\hat{\alpha} = \sum_{i=1}^{N} \alpha^{(i)}$, then we can lower-bound the optimisation problem defined in Equation (6.21) with the following linear programming problem:

$$\begin{aligned} \min & \left(\hat{\alpha} + \boldsymbol{\beta}^T \hat{\mathbf{r}} \right) \\ \text{Subject to:} \quad & A_{\hat{r}} \hat{\mathbf{r}} + A_z \mathbf{z} \leq b \\ & \hat{\mathbf{r}}^L \leq \hat{\mathbf{r}} \leq \hat{\mathbf{r}}^U \\ & \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U. \end{aligned} \tag{6.22}$$

As a consequence we have the following.

Proposition 14. *Let $\underline{\sigma}^2$ be the solution of the linear programming problem defined in Equation* (6.22)*. Then $\Sigma_T^L := \sigma_p^2 + \underline{\sigma}^2$ is a lower bound for the posterior variance $\bar{\Sigma}_{x,x}$ in T.*

Convergence of Variance Bound The convergence of the bounds computed on the variance to the actual values in hyper-rectangles $R \subseteq T$, with $\text{diam}(R) \to 0$, is an immediate consequence of Lemma 4, and proceeds similarly to what we have shown for the a-posteriori mean. In fact, we have that the objective function for the upper bound (that is, Proposition 13) is exact, and the over-approximation comes only from the feasible region of the optimisation problem. However, this is relaxed by using LBFs and UBFs that come directly from Proposition 1, so that their uniform convergence implies that the over-approximated feasible region converges to the actual one in the limit of $\text{diam}(R)$ shrinking to 0. Similarly, for the lower-bounding of the variance, the only difference comes from the fact that we use Proposition 1 also for the lower-bounding of the optimisation function. However, this will converge as well to the actual objective function. Thus, the exact solution of both optimisation problems converges uniformly to the actual values, for R small enough.

Remark 4. *Convergence, as shown above, relies on the fact that we compute exact solution to linear programming problems and to quadratic convex problems. Though this is achievable in theory, in practice, because of floating point arithmetic and computational time requirements, it is difficult at times to guarantee that. So, although the bounds derived in this chapter are formal, their software implementation in this form can lead to numerical errors that may add up, and become relevant depending on the precision required by a given application (e.g., if the GP is controlling a robot performing a delicate intervention in humans, where even millimetres could be fatal). In those cases it is possible to account for the additional error first by solving the dual programming problems instead of the primal ones, as those provide a worst-case solution at any point in time. Second, it is possible to use bounds for finite-precision arithmetic and propagate them through the bounds computed in Proposition 1, and down through the whole optimisation pipeline.*

6.4.3 Computation of Under-approximations

As discussed in Section 6.1, in order to obtain $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ it suffices to evaluate the GP posterior predictive distribution in any point of T. However, the closer $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ are to $\pi_{\min}(T)$ and $\pi_{\max}(T)$, respectively, the faster the branch and bound algorithm will converge (as per line 7 in Algorithm 1). Notice that, in solving the optimisation problems associated to μ^L_T, μ^U_T, Σ^L_T and Σ^U_T, we obtain four extrema points in T on which the GP assumes the optimal values for the a-posteriori mean and variance bounds. As these points belong to T and provide extreme points for the latent function, they are promising candidates for the evaluation of $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$. Specifically, in line 6 of Algorithm 1, we evaluate the GP posterior distribution on all four of the extrema and select the one that gives us the best bound among them.

6.5 Interpretability Analysis

We now consider how adversarial robustness can be used to compute quantitative interpretability metrics over the GP predictions. In fact, as discussed in Chapter 3 and highlighted by the works on adversarial examples [189], there is a clear connection between adversarial examples and quantified interpretability of a model output. That is, through computing a local adversarial example one can quantify the susceptibility of an image (or test point) features in producing consisted predictions.

As an illustrative example of a possible metric, we build on LIME [170], which is a black-box method for local interpretability analysis of ML models that works by performing local linear approximations of the ML model and by looking at the coefficients of these approximations. Namely, we generalise that metric to the non-linear computations for the specific case of GPs. Specifically, given a test point x^* and a $\gamma > 0$, consider the one-sided intervals $T^j_\gamma(x^*) = [x^*, x^* + \gamma e_j]$ (with e_j being the vector of 0s except for 1 at dimension j) and $T^j_{-\gamma}(x^*) = [x^* - \gamma e_j, x^*]$. We compute how much the maximum and minimum values can change over the one-sided intervals in both directions:

$$\boldsymbol{\Delta}^j_\gamma(x^*) = \left(\pi_{\max}(T^j_\gamma(x^*)) - \pi_{\max}(T^j_{-\gamma}(x^*))\right) \tag{6.23}$$

$$+ \left(\pi_{\min}(T^j_\gamma(x^*)) - \pi_{\min}(T^j_{-\gamma}(x^*))\right). \tag{6.24}$$

Intuitively, this provides a non-linear generalisation of numerical gradient estimation, which is close to the metric used in [170] as γ shrinks to 0. Interestingly, this allows us to take into account non-linear behaviour in the finite proximity of a test point, which would be ignored by a differential analysis of the prediction. Notice how the maximum and minimum can be over-approximated by using the optimisation framework introduced above in the case of GPs classification models.

The measure can also be used straightforwardly for the evaluation of the global behaviour of a given feature. In fact, while $\boldsymbol{\Delta}^j_\gamma(x^*)$ is local to a given x^*, following LIME, global interpretability information is obtained by averaging local results over M test points, i.e. by computing

$$\boldsymbol{\Delta}^j_\gamma = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Delta}^i_\gamma(x^{*,(i)}).$$

6.6 Computational Complexity

Proposition 8 implies that the bounds in Proposition 7 can be obtained in $\mathcal{O}(M)$, with M being the number of intervals the real line is being partitioned into (this should scale proportionally to $\frac{1}{\epsilon}$, where ϵ is the desired error tolerance, as discussed in the proof of Theorem 3). The computational complexity for computing μ^L_T, μ^U_T and Σ^U_T was discussed in Chapter 5. Concerning Σ^L_T, we have to solve $2N + 1$ linear programming problems, where N is the size of the training set. In practice, some of these problems need not be computed, which means that a looser bound is computed, but faster. Refining through branch and bound has a worst-case cost exponential in the number of dimensions of T. While the mean computation is straightforward, the

computations for the variance are more computationally involved. In practice, we find it beneficial to just update the mean bounds and refine those for the variance only if convergence does not happen after the first few thousand of iterations. This also implies that the regression case is much faster, as variance computation is not required in that setting.

Regarding the multi-class case, we have that in practice an increased computational cost comes from the fact that we need to discretise a multi-dimensional latent space. Similarly to what we have seen in the two-class case, we have that, in order to achieve convergence up to a value ϵ, a discretising grid of the order of $\frac{1}{\epsilon^m}$ needs to be used, which makes the bound explode very quickly with increasing m. Another practical issue due to using the multi-class bound is that, while in the two-class case the sigmoid function is interesting (that is, significantly different from either 0 or 1) only in a small interval of $\mathbb{R}$, the softmax function is dependent on relative values between components rather than their absolute values, and the use of a fixed grid is sub-optimal in this case. Instead, we proceed by building an actual grid only around the mode of the Gaussian posterior distribution at the test point x^*, and choosing the width that depends on the posterior variance around there. For the rest of the points in the space, in fact, the GP assigns probability almost 0 to any realisation, so that their relative importance to the bound vanishes, and we are still able to obtain a tight bound in this case as well.

In the experimental section, we will provide analysis for the actual running time of the bound computation.

6.7 Experimental Results

We employ the methods developed in this chapter to experimentally analyse the robustness of GP models in adversarial settings. We give results for three datasets: (i) Synthetic2D, the two-dimensional dataset that was introduced in Example 7; (ii) the SPAM dataset [46]; (iii) a two-class subset of the MNIST dataset [122] with classes 3 and 8 (i.e., MNIST38) and a three-class subset with classes 3, 5 and 8 (i.e., MNIST358).

Training For the Synthetic2D dataset we analyse the GP that was trained in Example 7. For the SPAM dataset we first standardise the data to zero mean and unit variance. Then, we perform a feature-reduction step by iteratively training an ℓ_1-penalised logistic regression classifier and discarding the least relevant features, up

until test set accuracy starts to diminish. This procedure leaves us with 11 features out of the initial 57. We then train a two-class classification GP over the training set, with a zero mean prior, a squared exponential kernel (employing MLE for estimating the kernel hyper-parameters) and use the probit likelihood. The GP thus computed achieves a test set accuracy of around 93%.

After sub-sampling the images to 14×14 pixels[2], we use similar settings for the MNIST38 dataset, achieving a test set accuracy of around 98% while training on 1000 training samples. Finally, for MNIST358 we perform multi-class classification using the softmax likelihood function and training setting similar to that for MNIST38, obtaining a test set accuracy of around 93%.

Unless otherwise stated, the approximate posterior distribution is computed using the Laplace estimation method.

Analysis Settings We compute adversarial robustness in neighbourhoods of the form $T = [x^* - \gamma, x^* + \gamma]$ around a given point x^* and for $\gamma > 0$. Unless otherwise stated, we run the branch-and-bound algorithm until convergence up to an error threshold $\epsilon = 0.02$. Similarly to what we did in the experimental evaluation of Chapter 5, for MNIST38 we perform a feature-level analysis for scalability reasons. Namely, we restrict our analysis only to salient patches of each image. We employ SIFT [130] or use the relevant pixels corresponding to the shortest GP length-scales in order to define those salient patches. We note that any other method for feature selection from images can be used in the place of this.

6.7.1 Runtime analysis

We first empirically analyse the runtime of the branch-and-bound method proposed here on the MNIST38 dataset. Namely, we aim to asses how the CPU time required by branch-and-bound for the computation of $\pi_{\max}(T)$ is affected by (i) the size of the input set T; (ii) the desired error threshold ϵ and; (iii) the number of training points, N. All runtimes analysed below were obtained on a MacBook Pro with a 2.5 GHz Intel Core i7 processor and 16GB RAM running on macOS Mojave 10.14.6, by averaging the results for 50 test points randomly selected from MNIST38.

[2]This reduces the number of hyper-parameters that need to be estimated by MLE and increases the numerical stability of the GP, while achieving comparable accuracy.

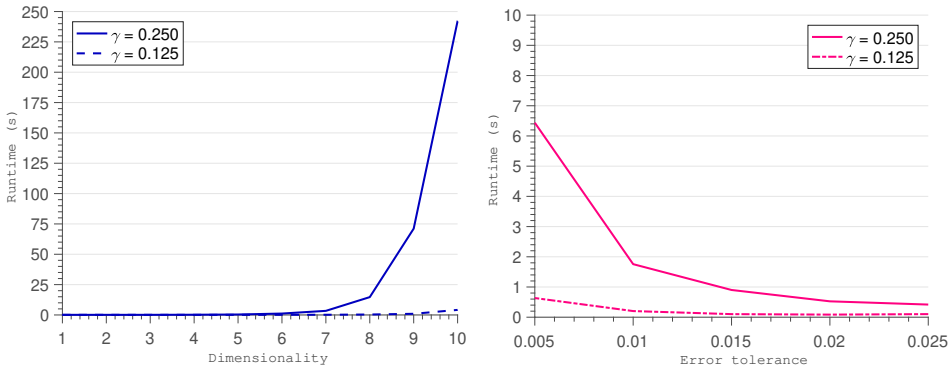


Figure 6.2: Average runtimes of Algorithm 1 to calculate $\pi_{\max}(T)$ up to a specified error tolerance ϵ for 50 test points randomly taken from MNIST38. **Left**: Average runtimes as we increase the number of input dimensions. **Right**: Average runtimes for different values of ϵ for a fixed number of dimensions $d = 5$.

Effect of Input Set Size In the left plot in Figure 6.2 we depict our algorithm running times for increasing number of input dimensions considered during computation (that is, the number of image pixels that can actually be changed, which defines the effective dimension of T). Namely, we vary the dimension of T from 1 (i.e., single-pixel variations) to 10, and analyse the results for $\gamma = 0.125$ and $\gamma = 0.25$. As discussed in Section 6.6, we have that the branch-and-bound algorithm has a worst-case computational time which is exponential in d, and we empirically observe that for $\gamma = 0.25$ the computational time starts to increase quickly already for $d = 10$. It is interesting to note also the magnitude of the effect that γ has on the running time. Simply halving γ reduces the computational time required for $d = 10$ from about 4 minutes to just about 4 seconds.

Effect of Error Tolerance In the right plot in Figure 6.2 we depict our algorithm running times against an increasing error tolerance ϵ that varies from 0.005 to 0.025, for $d = 5$. Of course, we obtain that looser requirements (i.e. greater values for ϵ) require less computing time. Interestingly, also in this case we observe an exponential (decaying) trend for the empirical computational time.

Effect of Number of Training Points We analyse the effect that the number of training samples has on the computational runtime of our method in Figure 6.3. The plot is obtained using $\gamma = 0.1$ and the 5 most important pixels as selected by SIFT. The required time grows polinomially with N up until branch-and-boudn refinement is not needed, after that the grows takes on an exponential rhythm.

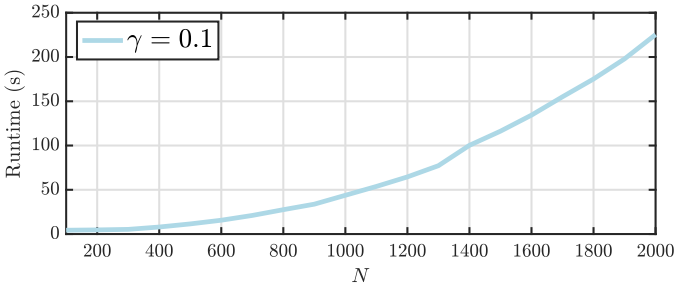


Figure 6.3: Average runtimes of Algorithm 1 on 50 MNIST test images with respect to the number of samples, N, used at training time.

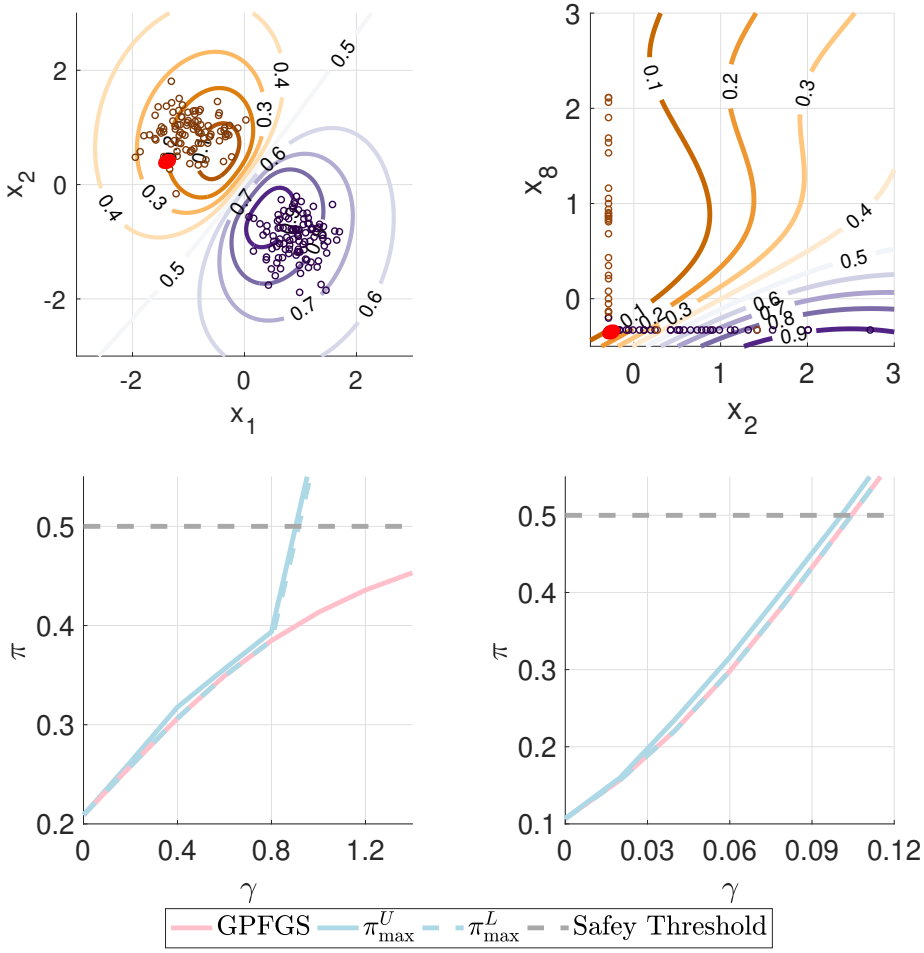


Figure 6.4: **First row**: Contour plot and test points for Synthetic2D (left); projected contour plot and test points for 2 dimensions of SPAM (right, dimensions 2 and 8 as selected by ℓ_1-penalised logistic regression); red dots mark selected test points. **Second row**: Safety analysis for the two selected test point. Shown are the upper and lower bounds on $\pi_{\max}(T)$ (solid and dashed blue curves) and the GPFGS adversarial attack (pink curve).

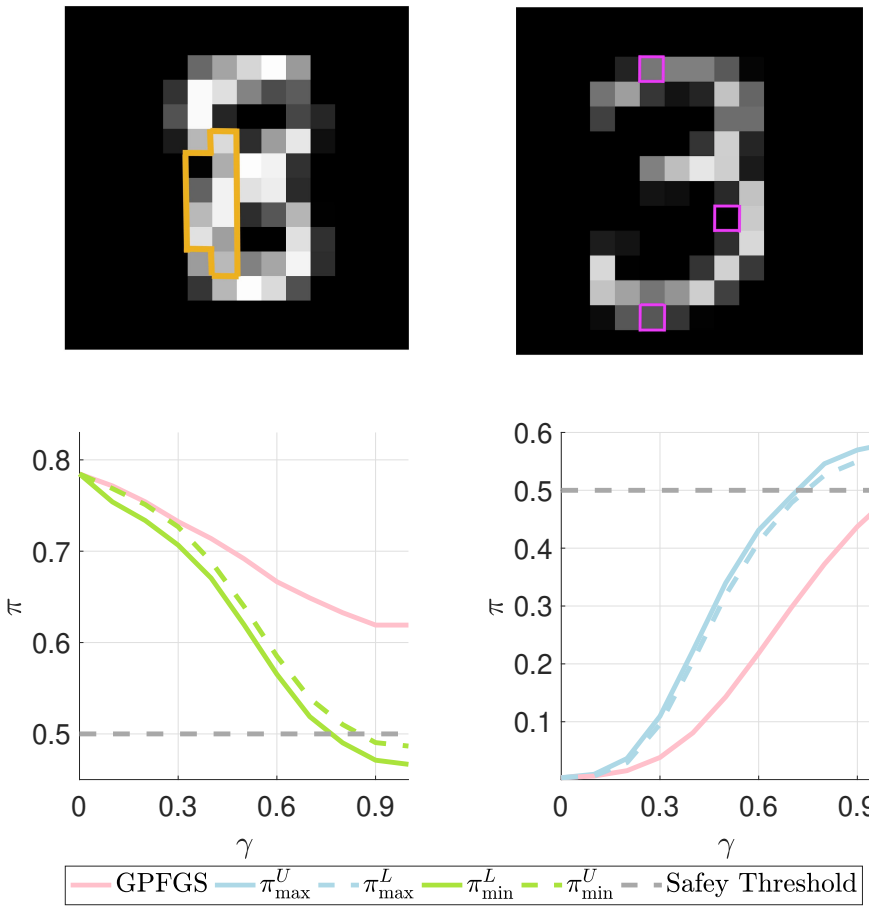


Figure 6.5: **First row**: Sample of 8 from MNIST38 along with 10 pixels selected by SIFT (left) and sample of 3 from MNIST38 along with the 3 pixels that have the shortest lengthscales after GPC training (right). **Second row**: Safety analysis for the two images. Shown are the upper and lower bounds for $\epsilon = 0.02$ on either $\pi_{\max}(T)$ or $\pi_{\min}(T)$ (solid and dashed blue respectively green curves) and the GPFGS adversarial attack (pink curve).

6.7.2 Adversarial Local Safety

We depict the local adversarial safety results for four points selected from the Synthethic2D, SPAM, and MNIST38 datasets in Figures 6.4 and 6.5. To this end, we set $T \subseteq \mathbb{R}^d$ to be a ℓ_∞ $\gamma-$ball around the chosen test point and iteratively increase γ (x-axis in the second row plots), checking whether there are adversarial examples in T. Namely, if the point is originally assigned to class 1 (respectively class 2) we check whether the minimum classification probability in T is below the decision boundary threshold, that is, if $\pi_{\min}(T) < 0.5$ (resp. $\pi_{\max}(T) > 0.5$). We compare the values provided by our method (blue solid and dashed line for class 2, green solid and dashed line for class 1) with GPFGS (a gradient-based method for attacking GPs mean pre-

diction [85], pink line curve in the plot). Naturally, as γ increases, the neighborhood region T becomes larger, hence the confidence for the initial class can decrease. Interestingly, while our method succeeds in finding adversarial examples in all cases shown (i.e. both the lower and upper bound on the computed quantity cross the decision boundary), the heuristic attack fails to find adversarial examples in the Synthetic2D and the MNIST38 case. This happens as GPFGS builds on linear approximations of the GP posterior distribution function, hence failing to find solutions when its non-linearities are significant. In particular, near the point selected for the Synthetic2D dataset (red dot in the contour plot) the gradient of the GP points away from the decision boundary. Hence, no matter what value γ takes, GPFGS will not increase above 0.5 in this case (pink line of the bottom-left plot). On the other hand, for the SPAM dataset, the GP model is locally linear around the selected test point (red dot in top right contour plot). Interestingly, the MNIST38 examples (Figure 6.5) provide results analogous to those for Synthetic2D. While our method finds adversarial examples on both occasions, GPFGS fails to do so (even with $\gamma = 1.0$, which is the maximum region possible for normalised pixel values).

6.7.3 Adversarial Local Robustness

We evaluate the empirical distribution of the adversarial prediction ranges (see Definition 6) on 50 randomly selected test points for each of the three datasets considered. That is, given T, we compute $\delta = \pi_{\max}(T) - \pi_{\min}(T)$. Notice that a smaller value of δ implies a more robust model. In particular, we analyse how the GP model robustness is affected by the training procedure used. We compare the robustness obtained when using either the Laplace or the Expectation Propagation (EP) posterior approximations technique. Further, we investigate the influence of the number of marginal likelihood evaluations (epochs) performed during hyper-parameter optimisation on robustness.

Results are depicted in Figure 6.6, for 10, 40 and 100 hyper-parameter optimisation epochs. Note that the analyses for the MNIST38 samples are restricted only to the most influential SIFT features, and thus δ values for MNIST38 are smaller in magnitude than for the other two datasets (for which all the input variables are simultaneously changed). Interestingly, this empirical analysis demonstrates that GPs trained with EP are consistently more robust than those trained using Laplace. In fact, for both Synthetic2D and MNIST38, EP yields a model about 5 times more robust than Laplace. For SPAM, the difference in robustness is the least pronounced. While Laplace approximation works by local approximations, EP calibrates mean and

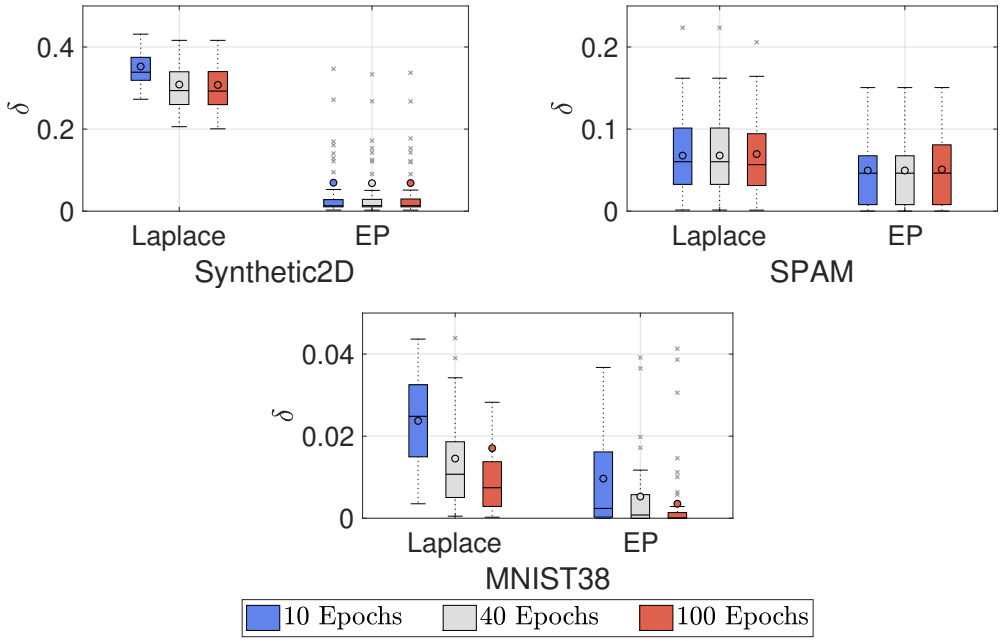


Figure 6.6: Boxplots for the distribution of robustness on the three datasets, comparing Laplace and EP approximation.

variance estimation by a global approach, which generally results in a more accurate approximation [167].

We compare Laplace and EP posterior approximations with that made by Hamiltonian Monte Carlo (HMC), that is, as in [143] we use HMC as a gold standard for Bayesian inference. The empirical distances found on the posterior approximation w.r.t. HMC are on average as follows (smaller values are better): (i) Synthetic2D - Laplace: 1.04, EP: 0.14; (ii) SPAM - Laplace: 0.35, EP: 0.32; (iii) MNIST38 - Laplace: 0.52, EP: 0.32. This shows a correlation between the robustness and the posterior approximation quality in the datasets considered. These results quantify and confirm for GPs that a more refined estimation of the posterior is beneficial for model adversarial robustness [27]. Interestingly, the values of δ decrease as the number of training epochs increases, thus robustness improves with training epochs. This is in contrast to what is observed in the deep learning literature [194]. More training in Bayesian settings may imply better calibration of the latent mean and variance function to the observed data.

6.7.4 Interpretability Analysis

Finally, we show how adversarial robustness can be used for interpretability analysis for GP models. We provide comparison with pixel-wise LIME [170], a model-agnostic

interpretability technique that relies on local linear approximations.

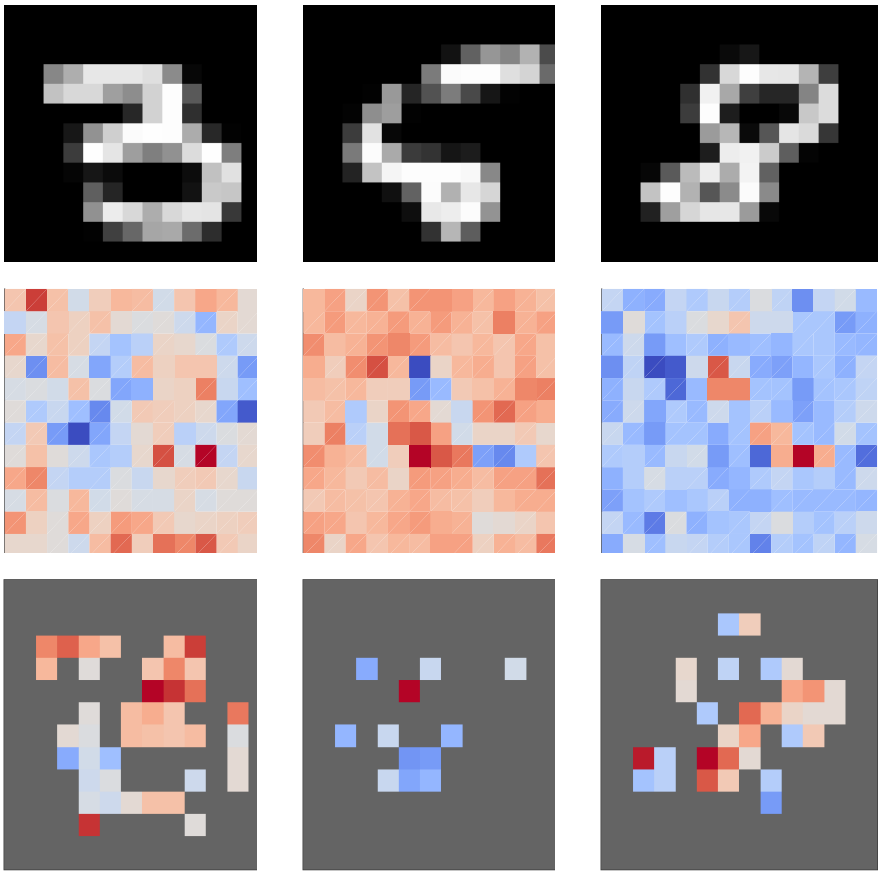


Figure 6.7: **First row**: Samples selected from MNIST358. **Second row**: Interpretability metric estimation using our method. **Third row**: Results obtained using LIME.

Local Interpretability for MNIST358 Figure 6.7 shows the results for three samples selected from MNIST358 (top row), with the heat maps depicting the results of our method (second row) and those for LIME (third row, greyed out pixels are marked as irrelevant by LIME). The colour gradient varies from red (positive impact, pixel value increase resulting in increased class probability of shown digit) to blue (negative impact, pixel value increase decreasing the class probability). For digit 3, our method obtains, for example, a contiguous blue patch on the left. Increasing the values of these pixels would modify the 3 into an 8. Indeed, when whitening the pixels of the blue patch, the class 3 probability assigned by the model decreases from 0.58 to 0.40. Similarly, for digit 5, our methods identify a blue patch that would change the 5 into an 8, and again the GP model indeed lowers its class 5 probability when the patch is whitened. Similarly, for digit 8, our method identifies a blue patch of 3

pixels towards the top left, which would turn it into something resembling digit 3 if whitened.

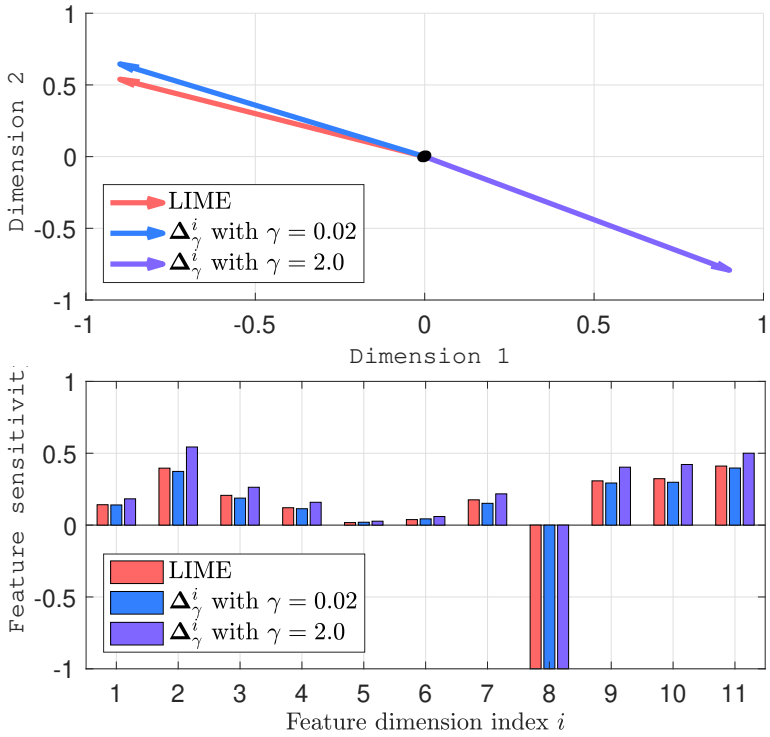


Figure 6.8: Global feature sensitivity analysed by LIME and our metric $\mathbf{\Delta}_\gamma^i$. All values normed to unit scale for better comparison. **Top:** Results for Synthetic2D dataset mapped out on plane. **Bottom:** Results for SPAM dataset.

Global Interpretability for the Binary Datasets We perform global interpretability analysis on the GP models trained on the Synthetic2D and SPAM datasets, using 50 random test points. The results are shown in Figure 6.8. For Synthetic2D (top row), LIME suggests that a higher probability of belonging to class 1 (depicted as the direction of the arrow in the plot) corresponds to lower values along dimension 1 and higher values along dimension 2. As can be seen in the corresponding contour plot in Figure 6.4 (top left), the exact opposite is true, however. LIME, as it is built on linearity approximations, fails to take into account of the global behaviour of the GP. When using a small value of γ our approach obtains similar results to LIME. However, with $\gamma = 2.0$ the global relationship between input and output values is correctly captured. For SPAM, on the other hand (Figure 6.8, bottom), due to linearity of the dataset and GP, a local analysis correctly reflects the global picture.

6.8 Summary

We presented a set of methods for computing, for any compact set of input points, adversarial robustness guarantees over regression and classification GP models. In particular, we have developed a branch-and-bound scheme that provides upper and lower bounds to the quantities needed for the analysis, and proved that it converges in finitely many steps to a value ϵ-close to the actual one, for any $\epsilon > 0$ selected a-priori.

In order to do so, we have found it necessary to extend the optimisation framework that we had previously developed in Chapter 5 for the computation of upper and lower bounds on the a-posteriori variance, and to provide guarantees of convergence of the lower and upper bounds computed using these techniques. We have then employed our method for the analysis of three datasets, providing results for adversarial robustness, bounds over the predictive posterior distribution and local/global interpretability analysis. One of the results observed in the experimental sections relates the robustness of a model with the "quality" of the posterior distribution approximation, and with the hyper-parameters calibration. Interestingly, these results suggest the existence of a positive correlation between adversarial robustness and the quality of fitness of a Bayesian model. This is completely opposite to what it is usually observed in frequentist approaches to learning, for example, in deep neural networks, where better accuracy was empirically observed to imply worst adversarial robustness [187].

We have performed further analysis of this trade-off (or lack thereof) in Bayesian neural network settings, by using statistical measures of adversarial global robustness in [121]. Interestingly, the cited work empirically confirms the preliminary results discussed in this chapter in large scale experiments with BNNs, where we have observed the existence of a positive relationship between accuracy and robustness in adversarial settings for BNNs. In follow on work [26], we have investigated the reasons behind this behaviour, and developed a plausible theoretical framework for the explanation of the behaviour for BNNs. Interestingly, this relies on an over-parameterisation limit for BNNs. As GPs can be seen as over-parameterised BNNs, that work could provide an interpretation for the behaviour that we observe in this chapter, and thus a theoretical confirmation for the results that we observe.

One of the main limitations of the approach presented in this chapter is related to its computational time. Fundamentally, the problem that we are trying to solve is a non-linear optimisation problem. From the universal theorem, we know that GPs

can approximate any smooth function, so that the problem of optimising a general GP is equivalent to that of optimising a general smooth function and is thus NP. As such, the exponential computational time requirement of the algorithm is to be expected. A possible mitigation technique, which was not explored in this chapter, is that of verifying sparse GPs. In fact, as seen in Section 6.6, the computational time for verification is strongly dependent on the number of training samples that we use, so that reducing their effective numbers implies a strong reduction in the computational requirement for convergence.

Finally, notice that the methods developed in this chapter for classification are strongly dependent on the posterior Gaussian approximation assumption. It is easy to see how the propositions presented can be generalised to the case in which a mixture of Gaussian distributions is employed. However, if different classes of approximating distributions are to be employed, new bounding propositions, specific to them, need to be developed.

Chapter 7

Robustness of Physiological Models for Affective Computing

Contents

In this chapter we investigate the behaviour of our methods developed for the computation of adversarial robustness of GPs in affective recognition problems. In fact, in directly dealing with the mental health of clinical and sub-clinical populations, the safety and the robustness of models learned in these setting is of paramount importance if those models are to be deployed in the wild. Furthermore, because of the necessity for human and expert participation in data collection experiments, affective datasets are often small in size, sparsely labelled and hence far away in

practice from the big data assumption necessary for end-to-end learning [44]. For instance, the three datasets introduced in Section 3.3 are composed of a number of overall training points that range from 52 to 348, making deep learning impractical in these settings. As such, affective recognition problems provide us with the perfect real world testbed for learning and verification of GP models.

In this chapter, we first describe how competitive state-of-the-art classification models based on GPs can be developed for affective recognition from physiological signals (Section 7.1). In order to do so, we will design physiologically-based GP priors that build on top of physiologically justified assumptions about the data generating process. We argue how this will allow us to learn the GP model directly from the raw data, enabling end-to-end learning in the settings when just a few hundreds of data points are actually available. In Section 7.3 we then implement and compare GP-based models from valence and arousal recognition from Electrodermal-Activity (EDA) and from (Heart Rate Variability) HRV signals, in the three datasets introduced in Section 3.3. The topic of Section 7.4 will thus be the verification and interpretability of the models derived by using the methods developed in Chapter 6. In order to do that, we will show how, thanks to the linearity of the posterior inference equations for GPs, the optimisation framework for the a-posteriori mean and variance can be straightforwardly extended in the case of non-null prior function Section 7.2. We then conclude the chapter with a summary of the overall observations made and a discussion on the results obtained and their physiological interpretations.

7.1 GPs for Affective Recognition

In settings in which large datasets are available, and especially in the case of BNNs, it is often argued in the literature that the use of an uninformative and flexible enough prior is sufficient for obtaining competitive modelling performances. As argued above and in Chapter 3, this is unfortunately not the case in affective recognition tasks, where datasets tend to count observations in the order of hundreds, which is several orders of magnitude less than datasets normally used in computer vision tasks. We thus begin this chapter with an investigation of whether and how prior design in Bayesian settings affects model performance, and use Gaussian processes for this purpose, as an illustrative argument in favour of a Bayesian principled approach.

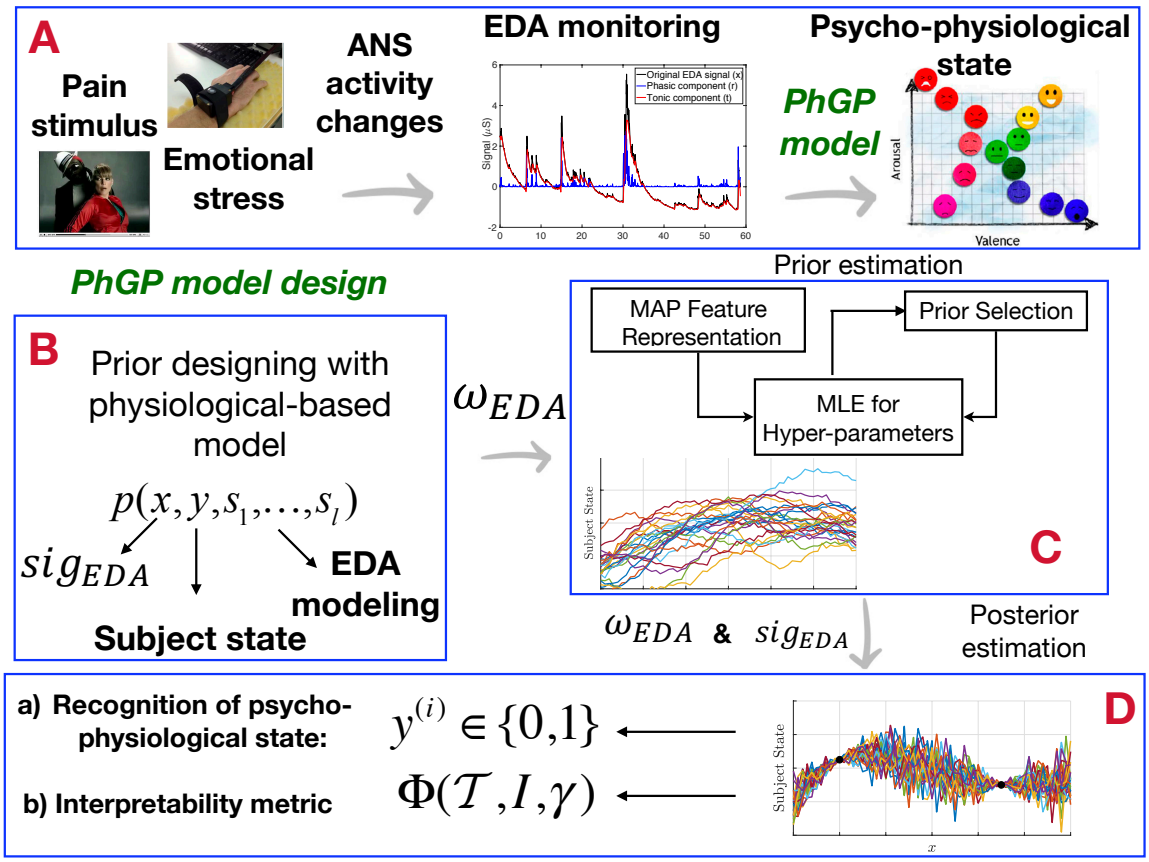


Figure 7.1: Pipeline of psycho-physiological state recognition with *PhGP* model.

7.1.1 Outline of the Approach

In particular, we aim at relying on prior information in order to build a learning system that can potentially take from the best of both worlds, that is, on the one hand incorporating previous problem-specific findings and physiological modelling in the form of a prior distribution over the GP model, while on the other hand working with raw physiological signals at inference time so that information not captured at the feature level can still be potentially extracted from the dataset in automatic fashion. This approach naturally flows from within the Bayesian learning framework, where the prior and the dataset information are simply merged by means of the inference formulas. For simplicity of reference, throughout the chapter, we denote a GP learned along these modelling lines as a *Physiologically-informed Gaussian Process* (*PhGP*) model.

A depiction of the complete prediction pipeline for the *PhGP* model is given in Figure 7.1, in the case of recognition of affective, psycho-physiological states from EDA recordings (block **A** of the plot). First, information from a physiologically-based

model of the physiological signals is encoded into a probabilistic generative model that captures their relationship with the raw input signal and the subject's affective state (block **B** in the plot). This is then used, together with a MAP estimation of the feature representation and an approximate MLE for the hyper-parameters, to define a Gaussian process prior over the *PhGP* model (block **C** in the plot). Posterior Bayesian inference is thus performed on top of *PhGP* to obtain a prediction for the subject's affective state $y^{(i)} \in \{0, 1\}$. Finally, we employ interpretability analysis on top of the *PhGP* to provide a quantitative explanation of the prediction made (block **D** in the plot). The specifics of each step necessary for the prior design are discussed in the remainder of this section.

7.1.2 Physiologically-informed Gaussian Process Prior

The *PhGP* model builds on Bayesian learning for embedding information from a physiological model (as those discussed in Section 3.3 for the case of EDA and HRV signals) directly into the training process. To do so, we simply aim at encoding the model as a functional distribution over the latent variable $f \in \mathbb{R}$ and feed it into the definition of the GP prior.

Specifically, we encode a physiological model as a set of unobservable sub-processes $\mathbf{s} = [s_1, \ldots, s_l]$. The assumption is that, given a subject's affective state, y, the process $\mathbf{s}$ gives rise to the observable physiological signal x according to a stochastic generative model of the form:

$$p(x, y, s_1, \ldots, s_l) \tag{7.1}$$

for an unknown density function p.[1] Ideally, the space in which the sub-processes contained in $\mathbf{s}$ are defined allows for a better understanding of the signal properties, and for the extraction of a set of relevant quantifiers, which we denote as $\omega(\mathbf{s}) = [\omega_1(\mathbf{s}), \ldots, \omega_n(\mathbf{s})] \in \mathbb{R}^n$. In this way, the relationship with the subject's affective state, y, is understood in terms of direct, physiological correlation.

In *PhGP*, we consider the feature vector $\omega(\mathbf{s})$ as the building block of the prior knowledge used to approximate the effect of $\mathbf{s}$ on the generation of the subject condition y in Equation (7.1) and employ a parametric approach to estimate it. Technically, all kinds of prior functions could be assumed. However, having justification or interpretation of why a complicated prior would be necessary is not easy. In our work, we

[1] In the case of EDA and HRV (discussed in Section 7.1.3), a physiological model aims to capture the relationship between the subject's condition and the SNS activity that gives rise to a variation of the signal properties.

investigate the suitability of a polynomial and a trigonometric parametric function, which are customarily used and implemented in standard tools for GP classification [168], as follows:

$$m_1(\mathbf{s}|\alpha) = \sum_{p=1}^{d}\sum_{j=1}^{n} \alpha_{pj}\omega_j(\mathbf{s})^p \tag{7.2}$$

$$m_2(\mathbf{s}|\alpha) = \sum_{p=1}^{d} \alpha_p^{(1)} \cos\left(\sum_{j=1}^{n} \alpha_{pj}^{(2)}\omega_j(\mathbf{s}) + \alpha_p^{(3)}\right), \tag{7.3}$$

where α is a vector of unknown hyper-parameters that adapt the shape of $m_j(\mathbf{s}|\alpha)$. Parameter d in Equation (7.2) is the degree of the polynomial function and we retrieve a constant, linear, quadratic and cubic function respectively for the values $d = 0, 1, 2, 3$. In Equation (7.3), instead, d represents the number of projected cosine components .

We then observe that, by using the probabilistic relationship that exists between x and $\mathbf{s}$, $m_j(\mathbf{s}|\alpha)$ can be used so to naturally induce a prior mean function over the GP. By marginalising over the random variable $\mathbf{s}$ we, in fact, obtain:

$$\mu(x|\alpha) = \int m_j(\mathbf{s}|\alpha)p(\mathbf{s}|x)d\mathbf{s}. \tag{7.4}$$

In general, this cannot be integrated for analytically, so in practice we employ a Monte Carlo approximation for its computation, as follows:

$$\mu(x|\alpha) \approx \sum_{i=1}^{M} m_j(\mathbf{s}_i|\alpha) \tag{7.5}$$

for M random samples of $\mathbf{s}$ from $p(\mathbf{s}|x)$. The prior function is then defined by the choice of the kernel function, for which we employ the squared-exponential kernel.

The mean and covariance resulting from the combination of Equation (7.4) and the squared-exponential kernel centres the *PhGP* prior around the model estimation provided by the physiologically-based generative model. The learning procedure for the *PhGP* model then follows the lines outlined in Section 3.2.2 for classification with GPs. In particular, for the a-posteriori mean of the *PhGP*, and in the case of the Laplace approximate posterior distribution, employing the inference equations of Property 6 we obtain:

$$\hat{\mu}(x^*) = \int m_j(\mathbf{s}|\alpha)p(\mathbf{s}|x^*)d\mathbf{s} + \mathbf{k}^{*T}K^{-1}\hat{\mathbf{f}}$$

where the solution provided by the physiological model is adapted by the raw signal data naturally following Bayes rule.

It is easy to see that a standard MLE for the hyper-parameters can be obtained for the *PhGP* model when using a pointwise estimation of the physiological process $\mathbf{s}$. To see that, let α denote the vector of hyper-parameters for the prior mean, and β denote the vector of hyper-parameters for the kernel function. Marginalising the latent variable, making explicit the dependence of the posterior distribution on the hyper-parameters and applying the inference formulas, we obtain that for the case of the Laplace approximation the marginal log-likelihood of the GP is given by:

$$\log p(\mathbf{y}|\mathcal{D},\alpha,\beta) = \log \int p(\mathbf{y}|f(\mathbf{x}))p(f(\mathbf{x})|\mathcal{D},\alpha,\beta)df(\mathbf{x}) = \\ -\frac{1}{2}\hat{\mathbf{f}}^T K^{-1}\hat{\mathbf{f}} + \log p(\mathbf{y}|f(\mathbf{x})) - \frac{1}{2}\log|I + W^{\frac{1}{2}}KW^{\frac{1}{2}}|, \tag{7.6}$$

where K explicitly depends on β, while both $\hat{\mathbf{f}}$ and W implicitly depend both on α and β. Equation (7.6) can be directly optimised for the values of the hyper-parameters to select the modelling that best justify the training data $\mathcal{D}$, which can be shown to provide an approximation of the maximum likelihood estimation for α and β [167]. In order to do so in the case of *PhGP*, we can employ the standard gradient-based optimisation method employed for GP models, by additionally propagating the derivatives with respect to α through Equation (7.4). To do that, we proceed by approximating the derivative computation by considering the MAP solution for the sub-process $\mathbf{s}$, so that we approximate:

$$\mu(x|\alpha) = \int m_j(\mathbf{s}|\alpha)p(\mathbf{s}|x)d\mathbf{s} \approx m_j(\mathbf{s}_{\text{MAP}}|\alpha). \tag{7.7}$$

We thus have that $\frac{d\mu(x|\alpha)}{d\alpha} \approx \frac{dm_j(\mathbf{s}_{\text{MAP}}|\alpha)}{d\alpha}$. The latter is straightfroward to compute as $\mathbf{s}_{\text{MAP}}$ does not depend on any hyper-parameters and can be computed analytically for Equations (7.2)-(7.3).[2]

7.1.3 EDA and HRV Priors

We now give an explicit formulation for *PhGP* in the case of EDA and HRV modelling, by considering the modelling procedures discussed in Chapter 3, i.e. *cvxEDA* for the EDA and feature analysis method for the HRV model. Specifically, these are used to build the vector of quantifiers $\omega(\mathbf{s}) = [\omega_1(\mathbf{s}), \ldots, \omega_n(\mathbf{s})] \in \mathbb{R}^n$, which are then used as the building blocks of the prior distribution.

[2]Numerical differentiation methods need to be used in the general case.

In particular, for the EDA model, we encode the parameters of the model in Equation (3.9) in the stochastic generative model of the form described by Equation (7.1), where x is the EDA signal and the phasic, the tonic and the SMNA driver of the phasic components (r, t, p) are the sub-processes denoted as s_1, s_2, s_3, which explain the generation of x. We then use standard quantifiers for x, t, r and p employed in the literature so as to form a set of features for the EDA-related component of the vector ω, i.e. $\omega_{\text{EDA}}(\mathbf{s})$:

- $\omega_{r,p}(\mathbf{s})$: Quantifiers from p and r: the number of significant phasic driver peaks ($nSCR$), the sum of SCR amplitudes ($SumAmpSCR$), the maximum value of SCR amplitudes ($MaxAmpSCR$), the mean value of phasic activity ($PhasicMean$) and the standard deviation of phasic activity ($PhasicStd$) [81].

- $\omega_t(\mathbf{s})$: Quantifiers from t: mean of tonic activity ($TonicMean$), standard deviation of tonic activity ($TonicStd$) [81].

- $\omega_x(\mathbf{s})$: Quantifiers from x: $EDASymp$, which is highly correlated to the activity of the sympathetic nervous system and is obtained by integrating the spectrum of x within the $(0.045 - 0.25Hz)$ frequency band [71].

On the other hand, concerning the HRV signal, we model the sympathetic and parasympathetic interaction by using time and frequency analysis methods discussed in Section 3.3. Namely, through the time domain analysis of HRV we will compute first and second order statistical moments, which we denote with μ_{HRV} and σ^2_{HRV}. By computing its frequency spectrum we then quantify the low-frequency (LF) and high-frequency (HF) indeces for HRV, as well as their ratio [1]. Hence, we define the HRV-related component of the vector ω, i.e. $\omega_{\text{HRV}}(\mathbf{s})$, as:

$$\omega_{\text{HRV}} = \left[\mu_{\text{HRV}}, \sigma^2_{\text{HRV}}, LF_{HRV}, HF_{HRV}, \frac{LF_{HRV}}{HF_{HRV}}\right]$$

Finally, for the case of arousal recognition and when employing both the EDA and HRV signal, we implement into the prior also a feature that quantifies the interaction between HRV and EDA signals. In fact, the correlation between the high-frequency power of the HRV with the parasympathetic activity and the EDA spectral power with the sympathetic outflow can be used for the evaluation of the sympathovagal balance [71, 72], which we denote as ω_{HE}.

7.2 Extension of Optimisation Framework for Non-null Prior Mean

The optimisation framework that we have introduced in Chapter 6 for the computation of adversarial robustness for GP was presented for the case in which the prior mean was assumed to be null (or constant). As discussed in Remark 1, while in the regression case we can assume a null mean without loss of generality, the same does not apply to classification models. In this section, we see how the formula for adversarial robustness can be extended to the case of non-null prior in the case of GP classification, and in particular we discuss the case of *PhGP* modelling.

By looking at the inference equations for the a-posteriori mean and variance[3], we notice that, while the functional form of the a-posteriori variance is not affected by a modification of the mean prior, the effect on the a-posteriori mean is two fold. First, the kernel matrix multiplication is centred around the prior mean computed on the training vector $\mathbf{x}$, which simply translates into a different definition for the constant vector $\mathbf{t}$ of Equation (5.8), so that this does not fundamentally alter the way in which computations are performed in Section 6.4. The second effect is linear on the prior function computed on the test point x^*, so that $\mu(x^*)$ is simply added to the computation of the a-posteriori mean. As such, in order to compute bounds in the case of a non-null prior mean function μ it suffices to compute a lower and upper bound over the a-priori mean function, that is, $\mu_T^{L,\text{pr}}$ and $\mu_T^{U,\text{pr}}$ such that:

$$\mu_T^{L,\text{pr}} \leq \min_{x \in T} \mu(x) \qquad \mu_T^{U,\text{pr}} \geq \max_{x \in T} \mu(x).$$

How to compute suitable values for $\mu_T^{L,\text{pr}}$ and $\mu_T^{U,\text{pr}}$ is a problem that depends, of course, on the exact form of the prior function used. In general, for prior mean functions that can be written down analytically, a variance of the methods discussed in Proposition 1 of Chapter 5 can be used, though in the general case one might have to resort to numerical optimisation methods if smoothness assumptions are not satisfied. The bounding problem is actually quite simple for the polynomial and trigonometric functions introduced in Equations (7.2) and (7.3), and can be solved by means of interval bound propagation techniques. In particular, in the polynomial case, the overall solution can be written down in closed from.

[3]Both in the case of Laplace approximation (see Property 6) and in the case of EP approximation (see Property 7).

Proposition 15 (Bound for Polynomial Function)**.** *Consider the following polynomial function:*

$$\mu(x) = \sum_{p=1}^{d} \sum_{j=1}^{n} \alpha_{pj} x_j^p$$

and let $T = [x^L, x^U] \subset \mathbb{R}^d$ *be an hyper-rectangle in the input space. Define*

$$\left[\bar{x}_j^{L,p}, \bar{x}_j^{U,p}\right] = \begin{cases} [\min_{x \in \{x_j^L, x_j^U\}} x_j^p, \max_{x \in \{x_j^L, x_j^U\}} x_j^p] & \text{if } \alpha_{pj} \geq 0 \\ [\max_{x \in \{x_j^L, x_j^U\}} x_j^p, \min_{x \in \{x_j^L, x_j^U\}} x_j^p] & \text{otherwise} \end{cases}. \tag{7.8}$$

Let:

$$\mu_T^{L,pr} = \sum_{p=1}^{d} \sum_{j=1}^{m} \alpha_{pj} \bar{x}_j^{L,p}$$
$$\mu_T^{U,pr} = \sum_{p=1}^{d} \sum_{j=1}^{m} \alpha_{pj} \bar{x}_j^{U,p}$$

then it holds that:

$$\mu_T^{L,pr} \leq \mu(x) \leq \mu_T^{U,pr} \qquad \forall x \in T.$$

For the case of the trigonometric prior function similar bounds can be obtained, by further accounting for the periodicity of the cosine function in the computations performed in Equation (7.8).

The application of the proposition above, together with the methods developed in Chapter 6, allow us to formally bound the posterior predictive distribution (and hence compute adversarial robustness) of the GP in the case in which a polynomial prior function is used. This yields the following Corollary:

Corollary 2 (A-posteriori Bound for Polynomial Prior Mean)**.** *Consider a GP with prior mean function given by Equation* (7.4)*, and kernel* Σ *with a bounded decomposition* (φ, ψ, U)*. Let* T *be an axis-aligned hyper-rectangle of the input space* $\mathbb{R}^d$*. Define* $\mathbf{t} = \Sigma_{x,\mathbf{x}} K^{-1}(\hat{\mathbf{f}} - \mu(\mathbf{x}))$*, and let* μ_T^L*,* μ_T^U *be computed as for Proposition 2 for this choice of* $\mathbf{t}$*. Consider* $\mu_T^{L,pr}$ *and* $\mu_T^{U,pr}$ *as defined in Proposition 7.8. Then it follows that* $\bar{\mu}_T^L$ *and* $\bar{\mu}_T^U$ *defined as:*

$$\bar{\mu}_T^L = \mu_T^{L,pr} + \mu_T^L$$
$$\bar{\mu}_T^U = \mu_T^{U,pr} + \mu_U^L$$

are upper and lower bounds for the a-posteriori mean of the GP in T*.*

Notice that the result stated in the corollary just above does not yet suffice to perform computations in the case of *PhGP*. In fact, while the prior functions is, in that case, polynomial (or trigonometric) over the feature vector ω, it is not so with respect to the input variable x, to which ω is only probabilistically related. Approximations for this case are discussed in the following section.

7.2.1 Propagating Through the Feature Space

In the *PhGP* model, the prior mean is defined by marginalisation over the process $\mathbf{s}$. In general, this cannot be computed exactly, and in fact in Section 7.1.2 we relied on Monte Carlo sampling to approximate its computation. Unfortunately, this makes analytical optimisation of the resulting mean intractable. Similarly to what we have done in the case of the computation of hyper-parameters, in order to compute bounds on the prior mean we rely on a MAP estimation of $\mathbf{s}$, so that we obtain:

$$\mu(x|\alpha) \approx \mu_{\text{MAP}}(x) := m_1(\mathbf{s}_{\text{MAP}}|\alpha)$$

where $m_1(\mathbf{s}_i|\alpha)$ is defined as in Equation (7.2). The idea is that Proposition 15 can be applied to the function above, once the hyper-rectangle T over x is mapped into an hyper-rectangle over the feature space ω. We denote with $\omega(x) = [\omega_1(x), \ldots, \omega_n(x)]$ the feature vector computed on top of the MAP solution for $\mathbf{s}$ given an observed value for x. Then we have that the following holds.

Proposition 16 (Bound for *PhGP* MAP estimation)**.** *Let for each $i = 1, \ldots, n$:*

$$\omega_i^L \leq \min_{x \in T} \omega_i(x) \qquad \omega_i^U \geq \max_{x \in T} \omega_i(x). \tag{7.9}$$

Let μ_ω^L and μ_ω^U be lower and upper bounds for the polynomial function of Equation (7.2) in $\prod_{i=1}^n [\omega_i^L, \omega_i^U]$ computed as for Proposition 15. Then it follows that:

$$\mu_\omega^L \leq \mu_{MAP}(x) \leq \mu_\omega^U \qquad \forall x \in T.$$

Hence, Proposition 16 guarantees that, given bounds on the feature vector ω, we can compute an overall bound on the prior mean function for *PhGP*, which we can then use for the computation of adversarial robustness. Unfortunately, still, the computation of the bounds on ω is not trivial in the general case. It is easy to see how rectangular bounds on the input space x can be propagated for time-domain and statistical features such as mean, standard deviation and min/max of the signals. For the general case, however, e.g., frequency-domain features, we instead rely on

numerical optimisation methods for the approximation of the bounds, and specifically on gradient-based methods. Notice that the upper and lower bounds obtained for *PhGP* can, in general, be only approximated for that reason, and are only as good as our approximation of the feature range is.

If we assume that the features extracted are smooth enough, then by relying on observations from numerical optimisation, we can expect the approximation to be good for input region T of small sizes, and to become looser as the region grows. In branch-and-bound settings, this implies that the approximation actually gets tighter as we explore the branch-and-bound tree by depth, which means that we still achieve convergence in the limit of infinitely many iterations. However, in practice, we lose the possibility of computing formal error bounds when the branch-and-bound is terminated after a finite number of iterations.

7.3 Experimental Model Validation

In this section, we first give detail of the experimental settings used and then provide comparison of the behaviour of *PhGP* in modelling affective recognition tasks with respect to GP trained with uninformative priors and Support Vector Machine (SVM) classification.

7.3.1 Experimental Settings

We perform our experiments on the three datasets that were introduced in Section 3.3, i.e., the CPT, DEAP and BVHP datasets for affective recognition from physiological signals. We pre-process the EDA signal in each training set by down-sampling it to 32 Hz and standardising to zero mean and unit standard deviation. We extract the heart rate signal from the ECG signal by using the Pan-Tompkins detection method for R-waves, and compute the HRV signals from that after smoothing of the obtained RR intervals. For the DEAP dataset, since the EDA of the 32 subjects were recorded by means of two different EDA acquisition systems, we select only the first 21 subjects, i.e., the largest group recorded with the same system, so as to avoid a bias in the features that was evident from a preliminary visual inspection of the signals.

We train and compare the results of 4 different models:

- *Raw-GP*: a GP learned directly from the raw physiological signal, with no prior physiological information is accessible by the model.

- *Feat-GP*: a GP learned solely from the feature vector ω extracted from the physiological signal, with the actual signal is not available to the model.
- *PhGP*: a GP with non-null prior which probabilistically combines the information from the physiological model and the raw signals, by embedding the former in the prior GP distribution, and by updating it by means of Bayesian inference directly computed on top of the raw input signal space.
- *SVM-RFE*: an SVM model learned with recursive feature elimination, as this provides a standard benchmark that is customarily applied for affective computing datasets of size comparable to those of the three we are here analysing [181].

The results reported are computed through a Leave-One-Subject-Out (LOSO) cross-validation procedure, so that the results and the models obtained are subject-independent, in terms of sensitivity, specificity and accuracy of predictions [220]. Namely, at each iteration of the LOSO validation scheme, the recognition model is trained using data from $M-1$ subjects (where M is the total number of subjects) and tested on the data from the left-out subject. This procedure is iterated M times.

7.3.2 Parametric Analysis of *PhGP* Prior Model

In Figure 7.2 we give results of a prior selection experiment on the CPT dataset. In particular, we analyse 6 different shapes of the prior function (i.e., null, constant, linear, quadratic, cubic and trigonometric), and three different feature vectors ω. Namely, with *Feature Set 1* we denote the results associated to only ω_{HRV}, that is, only the features extracted from the HRV signal. *Feature Set 2* denotes the results associated with the EDA-related features, that is, when using ω_{EDA}. Finally, *Feature Set 3* is used to represent the overall, combined results when utilising the full vector ω described in Section 7.1.3.

Interestingly, from the results we observe that the GP trained is able to obtain an accuracy significantly above 50% (i.e., that of a random classifier in this task) only when a non-trivial prior function is used, that is, only in the *PhGP* settings. Notice that the improvements are more marked for the cases in which HRV features are used (i.e., Feature Sets 1 and 2), which may be intuitively justified by observing that HRV dynamics tends to have a more markedly non-linear behaviour compared with that of EDA [158].

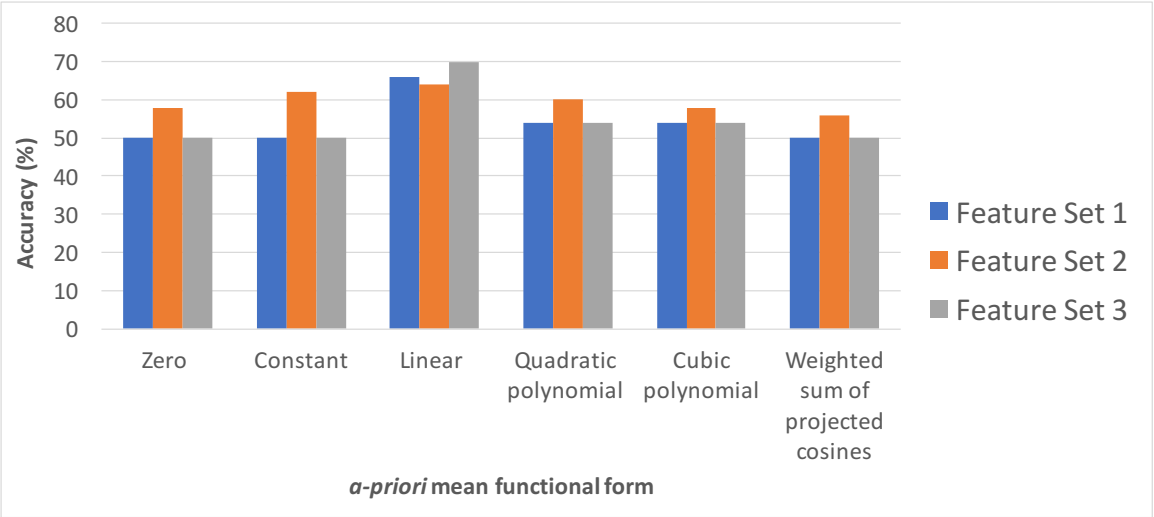


Figure 7.2: Classification results in terms of accuracy (%) using three sets of features with six different choices of mean prior function.

The highest LOSO accuracy ($\approx 70\%$) is achieved when *Feature Set 3* is used in combination with the linear prior function. Here, the results for the other feature sets also demonstrate that the linear function outperforms more complicated prior functions. This might be related with the risk of overfitting that comes from applying MLE for the estimation of the prior function hyper-parameter vector, α, whose size increases quickly for more flexible prior functions. In fact, MLE techniques have been observed in the literature to give rise to overfitting problems, similarly to those observed in frequentist learning settings. There is a significant increase in accuracy (approximately +20%) when the linear function is chosen for *Feature Set 3* compared to the case when a null *a-priori* mean function is considered. The recognition accuracy for *Feature Set 2* without the choice of prior function is 58%. Although this value is 2% higher compared to the form of weighted sum of projected cosines for the mean function, it is still 4% lower than the constant mean function and 6% lower than the linear function. The classification results for *Feature Set 1* also show a significant increase in recognition ($\approx 16\%$) when an appropriate prior function is chosen.

In Table 7.1 we list the the results for a prior selection experiment on the DEAP and BVHP datasets, in terms of LOSO sensitivity, specificity and accuracy. Notice that we do not give results for *PhGP* with zero and constant mean, as those correspond to the *Raw-GP* formulation, which is already included in the table. Comparative results of the performance of the *PhGP* model with *Raw-GP*, *Feat-GP* and *SVM-RFE* empirically demonstrate the advantages of relying both on physiological signal analysis and raw-signal information in terms of recognition performance. The

results reported in the table suggest that the *PhGP* model obtains higher accuracy for all choices of GP prior functions compared to the *Raw-GP* model both in DEAP and in the BVHP datasets. Though potentially having access to the same information (that is, the full raw signal), the *Raw-GP* model tends to overfit, while the *PhGP* methods benefits from the physiologically-informed prior in shaping its output distribution. Furthermore, *PhGP* provides results comparable to, and at times significantly better, than the *Feat-GP* model, which has access only to the features and not the raw signal. For example, with the linear prior function 3% and 10% lower overall accuracy is obtained when using *Feat-GP* compared to *PhGP*. Notice that *Feat-GP* significantly outperforms *Raw-GP* in the DEAP dataset (up to a 10% improvement), while the opposite is true for the BHVP dataset (up to an 8% improvement of *Raw-GP* compared to *Feat-GP*). While it is difficult to understand why that is the case, notice that *PhGP*, by drawing on from both models, is able to achieve competitive performance in both datasets, i.e., it successfully takes into account the information from the raw data and from the physiologically-based feature model. Moreover, for all the results obtained and reported in Table 7.1 we observe that choosing a non-trivial prior function leads to improved performance under LOSO validation, which confirms the results of Figure 7.2. Finally, we notice that, also in these settings, the linear prior gives more consistent performance across the two datasets.

In Figure 7.3 we analyse the performance of *PhGP* for different subsets of features taken from the EDA (introduced in Section 7.1.3) in the DEAP and BHVP datasets. We have that choosing the full set of features (i.e., ω_{EDA} and blue bars in the two plots) obtains the highest balanced performance between LOSO sensitivity and specificity of the prediction, and therefore highest accuracy compared to selecting subsets of features for the DEAP dataset. Although the feature subsets with ω_x and ω_t vectors result in higher sensitivity than ω_{EDA} in this dataset, the specificity is very low (53% for ω_x and 58% for ω_t). We do not observe any significant difference in this case for the BVHP dataset, for which all the models obtained comparable accuracy. This is due to the fact that*Raw-GP* (that is, when no physiological information is used for the definition of the prior) already obtains comparable accuracy in BVHP, so that *PhGP* can obtain similar performance independently of the prior function used.

7.3.3 Recognition results

We compare the *PhGP* model with SVM trained with recursive feature elimination [215] on the DEAP and BHVP datasets in Table 7.2. Building on the results from the previous section, we train the GP models using the linear form for the prior function

Dataset: DEAP									
	Raw-GP			*Feat-GP*			*PhGP*		
Prior	Sens.	Spec.	Acc.	Sens.	Spec.	Acc.	Sens.	Spec.	Acc.
Zero	84	47	65	73	56	64	–	–	–
Constant	67	63	65	72	52	64	–	–	–
Linear	66	67	68	69	87	78	81	82	81
Quadratic	74	70	72	77	80	78	68	78	73
Cubic	76	62	69	77	57	68	89	62	75
Trig.	73	68	70	73	58	65	92	51	72
Dataset: BVHP									
Zero	95	89	92	89	84	86	–	–	–
Constant	95	90	92	87	85	86	–	–	–
Linear	97	91	94	87	88	87	95	98	97
Quadratic	95	95	95	87	85	87	97	99	98
Cubic	93	93	95	87	85	86	95	99	98
Trig.	93	93	94	89	84	86	98	98	98

Table 7.1: Recognition results of the *Raw-GP*, *Feat-GP* and *PhGP* models considering different forms of functions for parametric modelling of the prior distribution for the two datasets. The values are expressed as percentages of sensitivity, specificity and accuracy of the performance of the recognition model.

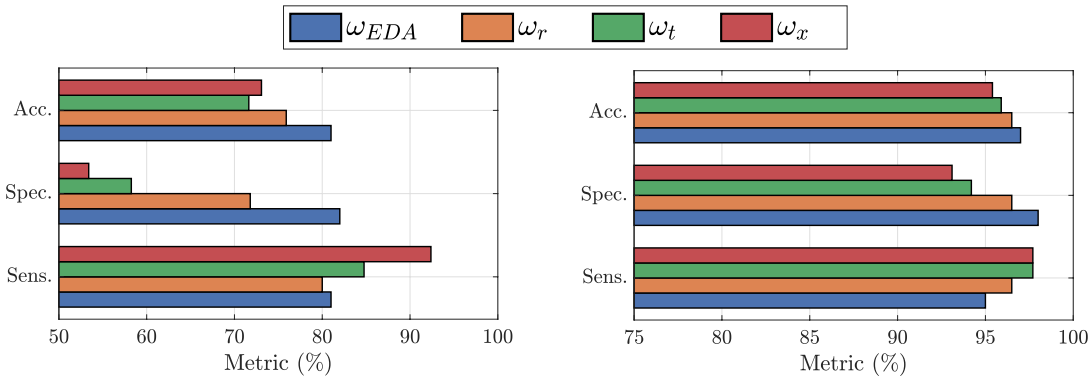


Figure 7.3: Comparative performance (sensitivity, specificity and accuracy) of *PhGP* with different subsets of features in the prior function for DEAP dataset (left) and BVHP dataset (right). Refer to Section 7.1.3 for the definition of each subset.

and, for the case of *PhGP*, using the full feature set for the vector ω. Overall, *PhGP* improves on the LOSO accuracy obtained by the SVM-RFE method by 17% and 7%, respectively, on the DEAP and BVHP raw datasets. It is interesting to note how all the GP models outperform the SVM-RFE method; in fact, the latter tend to overfit in these settings. Furthermore, using an MLE for the hyper-parameters of the prior in the GP settings, we also obtain a form of feature selection in the prior space, though in an approximate Bayesian fashion, which provides better generalisation properties.

The results of the *PhGP* model offer higher accuracy compared to a recent study performed in similar settings on the DEAP dataset, which obtained 71% accuracy [182] (similar to that obtained by SVM-RFE). Similarly to this study, previous studies have conducted experiments on the BVHP dataset with the aim of classifying the baseline and the highest pain level and validated their results with LOSO cross validation, achieving 74% [204, 129], 80% [128], 82% [105] and recently 85% [191] accuracy. Interesting, *PhGP* improves significantly on the accuracy of all of these methods. We finally notice that the higher recognition results obtained on the BVHP dataset compared to those for the DEAP dataset are probably due to the fact that pain detection can be considered an easier task than emotion detection, as confirmed in the literature [181, 205]. However, notice that we gain full benefit from that, especially when we consider the raw signal as a source of information as well. This suggests that many pain-related features that can be extracted from the physiological signals are not currently accounted for in the feature extraction methodologies that exist in the literature.

Model	Sensitivity (%)		Specificity (%)		Accuracy (%)	
	DEAP	BVHP	DEAP	BVHP	DEAP	BVHP
Raw-GP	74	95	70	95	72	95
Feat-GP	77	97	80	88	78	87
SVM with RFE	60	77	68	83	64	70
PhGP	81	95	82	98	81	97

Table 7.2: Comparison of the performance of *Raw-GP*, *Feat-GP* and *PhGP* models and the standard SVM algorithm embedded with RFE method for the two datasets.

7.4 Model Analysis Results

In the previous experimental section, we have empirically observed how the GP learning framework enabled us to learn competitive machine learning models on three

affective recognition tasks. This was enabled by their Bayesian formulation, which allowed us to learn models that were able to access information from raw signal data and from physiologically-based modelling, and combine these in a probabilistic fashion. In addition to obtaining good accuracy performance, for the purposes of this thesis, another advantage of GPs lies in the fact that, thanks to their analytical formulation, they are amenable of verification and interpretability analysis with the methods discussed in Chapter 6 and extended in Section 7.2.

7.4.1 Interpretability Results

We compute the interpretability index $\mathbf{\Delta}_\gamma^j(x^*)$ (introduced in Section 6.5) for each test point and for each model trained on the DEAP and BHVP datasets in a LOSO setting. The results of this analysis are plotted in the heat-maps in Figure 7.4 for the *Raw-GP*, *Feat-GP* and *PhGP* models. The vertical axis represents the subject index, which, since the analysis is done in LOSO settings, also means on a different GP, i.e. one that was learned on top of the remaining training data. The horizontal axis in the first and third column (corresponding to the *Raw-GP* and the *PhGP* model) represents the $\mathbf{\Delta}_\gamma^j$ value for each selected data patch from the input EDA signal. On the other hand, in the second column (i.e., that corresponding to the *Feat-GP* model), $\mathbf{\Delta}_\gamma^j$ is computed for each feature index in the ω_{EDA} vector, and the order in which they are put in the axis, is arbitrary. In all heat-maps the colour bar varies from blue, denoting the value 0 for $\mathbf{\Delta}_\gamma^j$, i.e. no effect on the predictive distribution, to red, denoting the value 1 for $\mathbf{\Delta}_\gamma^j$. Each colour bar is normalised with respect to the maximum value observed for the model, so that the maximum value of $\mathbf{\Delta}_\gamma^j$ in each plot is always 1.

It is interesting to note the consistency in the $\mathbf{\Delta}_\gamma^j$ values reported in each of these maps. Those refer to different models but learned in the same settings (only the training/test set split changes), so this highlights how the results obtained by interpretability analysis are in a sense qualitatively independent from the specific subjects, and shows that the model is learning features and patterns that are specific to the problem itself, rather than the particular subject involved. From the heat-map of the *Raw-GP* model in the DEAP dataset, we observe that the patches corresponding to the 48th-52th seconds of the whole 60s duration of the EDA acquisition has the highest contribution in recognition. On the other hand, the first 18 seconds and the 36th-42th seconds of the data show the least contribution in almost all of the subjects. This trend is different in the BVHP dataset, where the highest contribution is

correspondent to the only patch occurring around the 2th-3th second of the EDA signal. The heat-maps corresponding to the *Feat-GP* model show the high contribution of the $SumAmpSCR$ index in DEAP dataset whereas the features related to tonic activity ($Tonicmean$ and $Tonicstd$) are the most informative quantifiers for detecting the pain stimulus. Concerning the *PhGP* model, the patches of data corresponding to the highest value of $\mathbf{\Delta}_{\gamma}^{j}$ in the BVHP dataset are located at the first and the second second of the pain stimulus. These patches are possibly where the highest alterations in physiology due to pain stimulus are present.

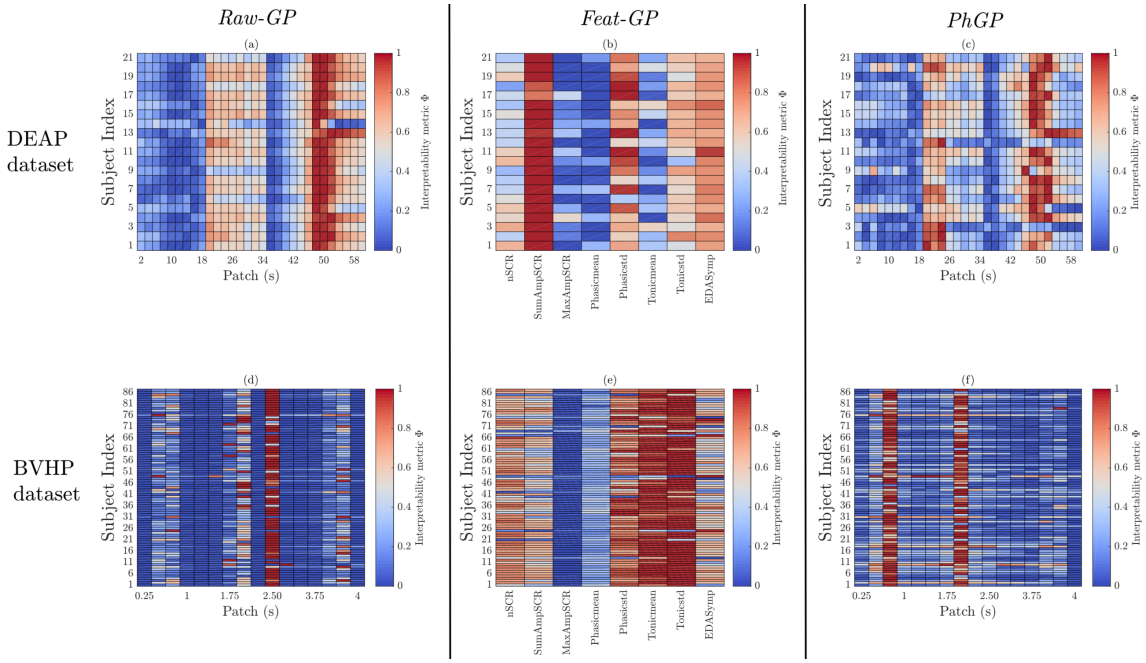


Figure 7.4: *Plots (a) and (d)*: heat-maps displaying the contribution of each data patch in DEAP/pain dataset for *Raw-GP* model. *Plots (b) and (e)*: heat-maps displaying contribution of each feature index for DEAP/pain dataset for *Feat-GP* model. *Plots (c) and (f)*: heat-maps displaying the contribution of each data patch for DEAP/BVHP dataset for *PhGP* model.

In Figure 7.5 we plot average results for the interpretability analysis over all the subjects for each model (blue line for average, and shaded blue area showing standard deviation), and across different values for γ (red line and red shaded area), ranging from 0 to 1. The similarity in the dynamical trends of the blue and red lines in the plot confirms and quantifies the consistency in interpretability analysis of all the models in both datasets regardless of the particular subject or the choice of γ value and subject.

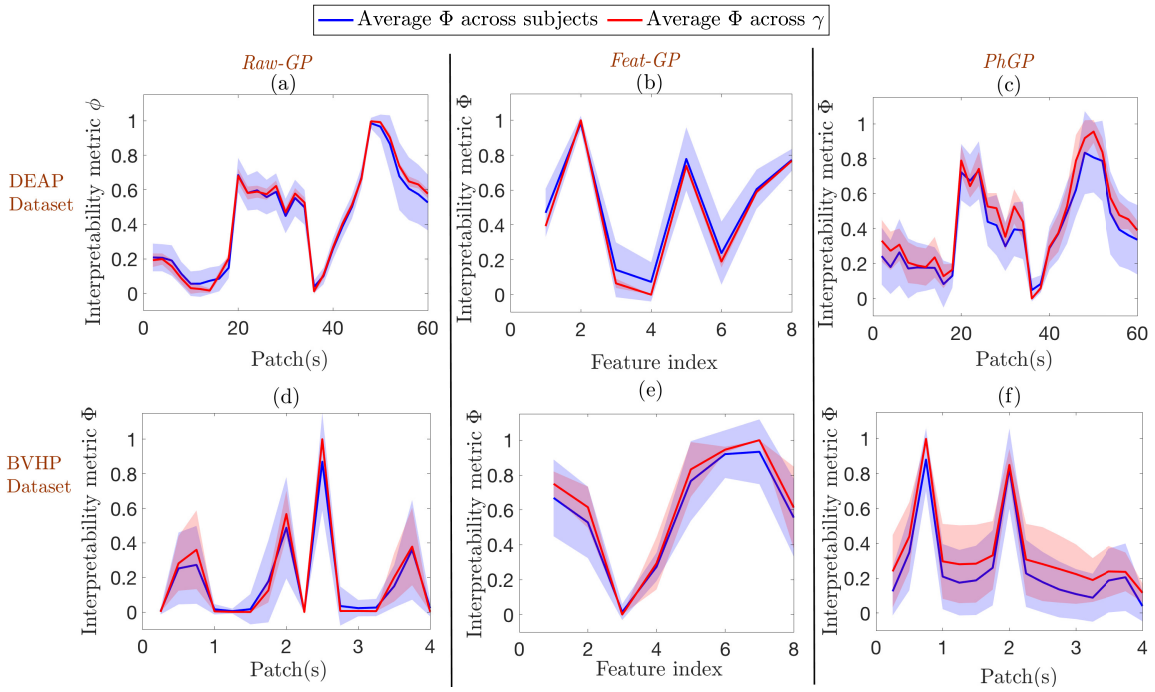


Figure 7.5: *Plots (a) and (d)*: show contribution of data patch for DEAP and BVHP datasets for *Raw-GP* model. *Plots (b) and (e)*: show contribution of each feature index for DEAP and BVHP datasets for *Feat-GP* model. *Plots (c) and (f)*: show contribution of each data patch for DEAP and BVHP datasets for *PhGP* model. The thick continuous blue and red lines indicate, respectively, the average of $\mathbf{\Delta}^j_\gamma$ metric across subjects and across γ values. The shaded blue and red areas indicate their respective standard deviation estimation.

7.4.2 Verification Results

We compute bounds on the minimum and maximum of the predictive posterior distribution on the *PhGP* model trained on the CPT dataset. We vary the maximum radius of perturbation allowed, γ, from 0 to 0.4, and allow for variation of the whole input space of the model. We notice, by inspection of the data, that, since we are standardising the data to zero mean and unit standard deviation (and not normalising them to $[0, 1]$) as it is customarily done for physiological signals, the gamma radius of 0.4 already covers almost entirely the empirical data manifold. For simplicity of representation, the line in the plots show the trend of the bounds only on 10 input points randomly sampled from the test set. We use an error tolerance $\epsilon = 0.01$ in the branch-and-bound. Recall that, since this is a two-class classification problems, to verify the behaviour of the model, it suffices to compare the bounds with respect to the Bayesian optimal decision threshold, i.e., 0.5.

We plot the results of these analysis in Figure 7.6; in the left plot we show the

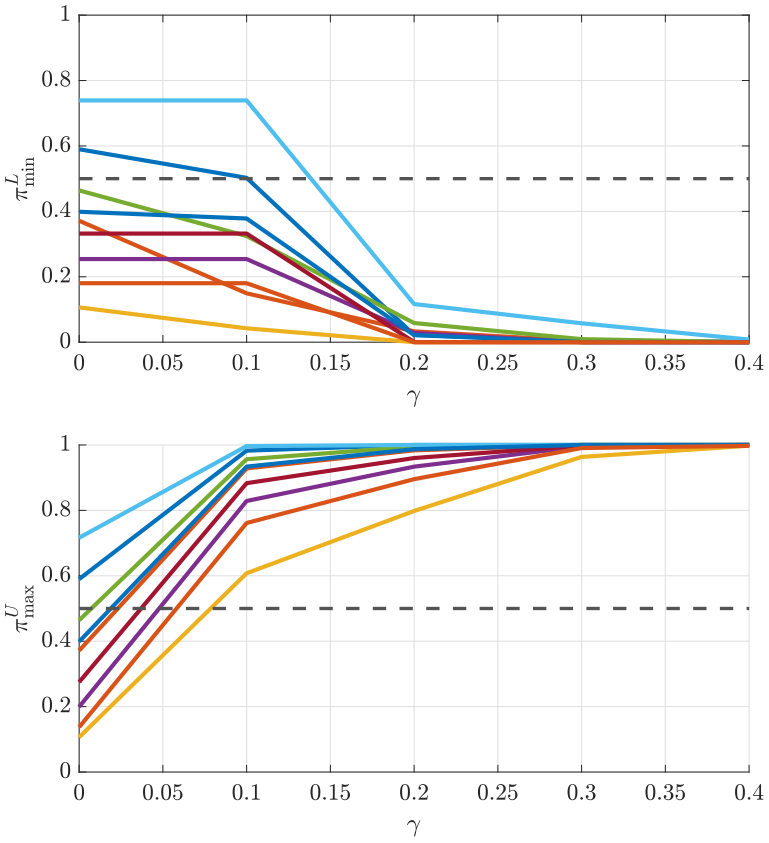


Figure 7.6: Computation of lower bound on minimum (left plot) and upper bound on maximum (right plot) of the predictive posterior distribution of *PhGP* on 10 test points randomly sampled from the CPT dataset. Verification for the Bayes optimal Classifier can be retrieved by comparing the solid lines with the decision threshold (dashed grey line).

lower bound on the minimum of the predictive probability, while the upper bound on the maximum is shown in the right plot. The decision threshold is depicted with a grey dashed line. The coloured solid lines, represents the values of the bounds obtained for each one of the 10 test points selected. The existence of an adversarial attack at given value for γ for a point which is initially classified in class 2 (that is the value at $\gamma = 0$ lies above the decision threshold) can be retrieved from the plots by checking whether $\pi^L_{\min}$ is below the threshold. Conversely for a point that starts from below the decision threshold we have to check whether at $\pi^U_{\min}$ at a given γ is above the threshold. Interestingly, all the points have a similar trend and the gradients of the bounds with respect to γ is qualitatively similar. Notice that, already for $\gamma = 0.2$ all the points selected are susceptible to adversarial attacks.

7.5 Summary

In this chapter we have applied GP models in three affective recognition dataset. First, we have shown how, by building on their Bayesian formulation, GPs provide us with a perfect learning framework for combining information from physiologically-based models and training data in the form of raw signals, which is particularly beneficial when only a small amount of labelled data is available at training time. Experimentally, we have also seen how GPs learned in these settings are able to obtain competitive prediction performance, while still relying on simple shallow models that can be interpreted. For this purpose, we have extended the framework for the computation of bounds on the posterior predictive distribution of a GP to the setting of non-null prior, and discussed approximation for the cases of the *PhGP* models. The so developed methodology was then employed to perform interpretbility analysis for the obtained model.

Further development of such techniques, and the definition of explainability metrics and extensions of the methodology derived here, can play a crucial role toward deployment of machine learning models in real-world affective recognition settings, as this would allow one to provide guarantees of model behaviour and/or explanations of why and under which condition a model might fail [170]. In fact, the interpretability of a model, and not its performances, is a key aspect to build trust of the clinical practitioners in automated affective tasks. Finally, notice that GPs are beneficial because of their data-efficient nature, but in situations where a large dataset is available similar techniques based on BNNs could be developed. One of the greatest obstacles in that direction is the fact that BNN priors are usually chosen to be uninformative because of the difficulty in interpreting their meaning. Even though, because of the "big data" assumption, one might find it unnecessary to build physiologically-informed priors in those settings, the formulation of such priors itself provides a simple justification for the model behaviour - being the model centred around the prior, and using the data only when the prior fails - which can be intuitively explained to practitioners.

Chapter 8

Conclusions

In this thesis, we have considered robustness of Bayesian inference with Gaussian process models under adversarial attack settings. Our investigation led us to discuss two different notions of robustness. The former, probabilistic adversarial robustness, captures the stochastic dynamics of the posterior model, and can, in fact, be seen as a rough generalisation of uncertainty measures to the adversarial prediction scenario. Then, in defining adversarial robustness, we have taken into account the overall decision of the model, which depends also on the loss function employed for Bayesian decision making, and can be seen as a generalisation of measures used in the frequentist learning settings (e.g. for deep neural networks). We have also discussed how, in the context of classification models, adversarial robustness is closely related to the computation of lower and upper bound ranges on the posterior predictive distribution of the Bayesian model.

The core methodology that was then developed throughout the thesis, is a general optimisation framework for posterior GP models, which is centred around the computation of lower and upper bounding functions for posterior mean and variance of the GP, along with other related and derived quantities. By relying on the Borell-TIS inequality and the Dudley entropy integral, we have seen how the optimisation framework allowed us to compute formal over-approximations of the probabilistic adversarial robustness of GPs. Regarding the computation of adversarial robustness, we have relied on a discretisation of the latent space to convert the problem to that of the optimisation of a set of Gaussian integrals, which again we formally bounded using our GP optimisation framework.

Utilised the central limit theorem, in order to employ the methods developed for GPs to the verification of infinitely-wide Bayesian neural networks. Thanks to the fact that inference for GPs is exact, this then allowed us to perform formal analysis of the behaviour of the BNN posterior under adversarial settings, which highlighted

some possible reasoning behind the mixed results observed in the literature for the use of uncertainty as a defense against adversarial attacks.

In further experiments on three benchmark datasets, we have found that adversarial robustness of the posterior GPs increases with a more refined training procedure (e.g., a better approximation or a more time-consuming MLE methods). This result suggests that exact posterior, along with a well-calibrated prior model, could provide a natural defence against adversarial examples in Bayesian settings. This is crucially different to what is observed in neural networks, where the more training is performed, the more the network becomes vulnerable to attack, and specific robust training techniques needed to be developed.

Finally, we have applied our techniques on three datasets to automate affective recognition from physiological signals. We have argued that, because of the small size typical of affective computing datasets and the absolute requirement of safety in situations of clinical relevance, GPs make a perfect modelling framework for this purpose. We have seen how carefully designing a prior on top of physiologically-based mathematical models allowed us to obtain competitive and even state-of-the-art performance on the three datasets analysed. We have then employed our techniques for GP verification to compute safety guarantees on top of a selection of these models, and performed formal interpretability analysis, which allowed us to obtain physiologically meaningful quantification and interpretation of the GP predictions.

8.1 Future Work

This thesis, analysed and discussed the first works, to the best of our knowledge, developed towards the computation of formal robustness guarantees for Bayesian models in adversarial settings. A number of the methods that we have discussed in previous chapters, and the preliminary results obtained in the case studies analysed here, have already been extended in works that I have co-authored during my DPhil.

An obvious research direction was that of extending the methods presented in Chapter 5 to the iterative prediction settings, which we achieved in [163], where we have also shown the application of such bounds for the synthesis of safe controllers. Furthermore, while the methods discussed in Chapter 5 apply to BNNs only in the infinite width limit, in [207] we have found that these computations could be performed for BNNs by relying on an optimisation framework akin to the one discussed in Chapter 6. The very preliminary observation we have made for the GP settings, about the correlation between the "quality" of the posterior computation and the

robustness against adversarial examples, we have later confirmed in large scale experiments on BNNs [121], and through the development of a theoretical framework [26]. Additionally, the methods developed in Chapter 5 for the computation of probabilistic adversarial robustness are formal, but suffer from scalability issues. Thus, in [27] we have developed a statistical model checking method for the computation of probabilistic robustness with a-priori statistical guarantees.

Several other directions for future research arise from the results of this thesis. We highlight some of these in the following paragraphs.

Adversarially Robust Training and Decisions for Bayesian models. While idealised Bayesian methods are provably robust to adversarial attacks [64, 26], in practice it might be infeasible or too computationally involved to arrive at a model close to the actual limit, robust behaviour. In these cases it would be beneficial to adapt the concept of certified robust learning [78] to the Bayesian settings. In order to do that in a probabilistic way, one can proceed by modifying the likelihood model so as to take into account not only the prediction with respect to the ground truth, but also the behaviour of the worst-case adversarial prediction. Of course, it is unreasonable to expect to obtain a methodology that provably converges for the general case, as it is difficult enough to obtain a model whose accuracy converges to the optimal one, since the training landscape is high-dimensional and non-linear for most problems of practical interest. However, by computing formal adversarial bounds such as those above, it is possible to provide certification for the behaviour of the objective function, and under smoothness assumption, for the improved robustness of models that are learned in a robust way, compared with models trained in a standard way. Interestingly, in Bayesian settings, robust training may also help the learning of the model and/or empirically improve the quality of our posterior approximation. In fact, since under the overparameterised assumption the posterior is provably robust, then pushing the learning towards robust posterior may have the effect of pushing it towards the true posterior.

An alternative approach involves working with the model loss function instead of the likelihood function. One can compute bounds on the posterior predictive distribution, or the posterior distribution itself, and use these to formulate a particular loss function that emphasises specific properties of the latent function, and not just the accuracy of the resulting model. In this way, training of the model proceeds in the standard way, while verification is used a-posteriori to guide the Bayesian decision making procedure.

Adversarially Robust Model Design. The preliminary results of Chapter 6, combined with the empirical and theoretical results we have discussed in [121] and [26], hint at the fact that when the prior model is correctly specified then the resulting Bayesian model will tend to be automatically robust against adversarial attacks. However, the question of how to correctly specify a prior model, in general, is not any simpler than coming up with a robust model. In the case of Gaussian processes, even though squared-exponential kernels provide us with a universal approximator, the question remains of how to select the hyper-parameters of the model, which is often done either qualitatively or by means of approximate inference methods on the available data. For the case of Bayesian neural networks, even though overparameterised networks with uninformative priors suffice as universal approximators, the practical computational requirements of these become prohibitive for real-world datasets (e.g. ImageNET). In Bayesian learning, the evidence framework was developed for principled probabilistic and data-driven model comparison [131]. We plan to employ the methodologies developed in this thesis to perform an extensive empirical and theoretical investigation of the suitability of the evidence measure for gauging the adversarial robustness of a model, and eventually building on its shortcoming with a similar approach to that discussed above for the modification of the likelihood function. Notice how methods for the modification of the likelihood functions and for the selection of the prior distribution could constitute building blocks of a future methodology for the synthesis of certifiable robust Bayesian models.

Verification of Hybrid and Deep Models. Scaling Bayesian models to be competitive compared to frequentist approaches has proven to be a rather difficult task. Recently models that try to take the best of both worlds have been developed, e.g. hybrid neural networks Gaussian process models [67, 24] and deep GPs [41]. These represent an effort to combine principled uncertainty quantification provided by Gaussian processes and the representational capabilities of deep learning. As the interest for these models increases in application scenarios [218, 31, 116], the question arises whether and how the techniques developed in this thesis can be employed in these settings. Intuitively, a probabilistic combination of the methods that we presented in [207] and [27] for tackling Bayesian neural networks with those discussed in Chapters 5-6 could provide a building block for their computation in hybrid models and deep networks. However it is not yet clear how to account for the probabilistic interface between them.

Physiologically inspired Neural Networks Priors. In Chapter 7 we have seen how, by relying on physiologically-based priors, GPs are able to obtain competitive and state-of-the-art results on affective computing tasks made up of relatively small datasets. We justified our choice of relying on GP models by the fact that they are amenable to verification thanks to the methods we had previously developed in Chapters 5-6. However, because of the recent development of similar methods for Bayesian neural networks [207], the question naturally arises as to whether something similar could be achieved using BNNs. In fact, GPs are fundamentally shallow models, and hence suffer from weak representational capabilities, which could be overcome by using BNNs, which would allow us to scale to more complicated tasks. Unfortunately the definition of meaningufl prior fucntions is a difficult task in BNN settings. In fact, for BNNs the prior is naturally defined over the weight space. The effect that this has on the latent space is non-linear and hence difficult to evaluate in practice. A qualitative understanding can be obtained by relying on the pioneering work of Neal [146], which shows how BNN priors can be evaluated in the limit by means of GPs. One could then consider employing the methods that we developed for GPs in Chapter 7 to the limiting GP, which represents the infinite-width limit of BNNs, leading to an approximate treatment of finite BNNs. In a second step, we could then look at whether interpretability metrics similar to that we discussed for GPs in Chapter 6 could be derived for BNNs, and how these can be employed for reverse-engineering specific prior functions.

8.2 Outlook

In this thesis we have investigated the problem of robustness for Gaussian processes in adversarial settings, providing the first of such treatments within the Bayesian learning paradigm. This has allowed us to experimentally find several interesting properties of Bayesian models, whose adversarial behaviour we have shown to be fundamentally different than that of the frequentist methods. One such is their tendency to be naturally robust to adversarial attacks when the model is accurately specified.

Compared to standard applications of (probabilistic) verification, and even to the verification of deterministic neural networks, analysis of adversarial robustness of Bayesian models is a new and unexplored field. As the search for robust machine learning models continues, we hope that the preliminary results and the methodologies discussed in this thesis will provide the motivation and the initial means to further investigate the many fascinating properties of Bayesian models.

Bibliography

[1] U Rajendra Acharya, K Paul Joseph, Natarajan Kannathal, Choo Min Lim, and Jasjit S Suri. Heart rate variability: a review. *Medical and biological engineering and computing*, 44(12):1031–1051, 2006.

[2] Robert J Adler. *The geometry of random fields*, volume 62. Siam, 1981.

[3] Robert J Adler and Jonathan E Taylor. *Random fields and geometry*. Springer Science & Business Media, 2009.

[4] Emad Alsuwat, Hatim Alsuwat, John Rose, Marco Valtorta, and Csilla Farkas. Detecting adversarial attacks in the context of bayesian networks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 3–22. Springer, 2019.

[5] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, and Mirco Marchetti. On the effectiveness of machine and deep learning for cyber security. In *2018 10th International Conference on Cyber Conflict (CyCon)*, pages 371–390. IEEE, 2018.

[6] Adelchi Azzalini and Marc G Genton. Robust likelihood methods based on the skew-t and related distributions. *International Statistical Review*, 76(1):106–129, 2008.

[7] Dominik R Bach and Karl J Friston. Model-based analysis of skin conductance responses: Towards causal models in psychophysiology. *Psychophysiology*, 50(1):15–22, 2013.

[8] V Balakrishnan, S Boyd, and S Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *International Journal of Robust and Nonlinear Control*, 1(4):295–317, 1991.

[9] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

[10] Theodore P Beauchaine and Julian F Thayer. Heart rate variability as a transdiagnostic biomarker of psychopathology. *International Journal of Psychophysiology*, 98(2):338–350, 2015.

[11] Artur Bekasov and Iain Murray. Bayesian adversarial spheres: Bayesian inference and adversarial examples in a noiseless setting. *arXiv preprint arXiv:1811.12335*, 2018.

[12] James O Berger. Robust bayesian analysis: sensitivity to the prior. *Journal of statistical planning and inference*, 25(3):303–328, 1990.

[13] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

[14] Samir Bhatt, Ewan Cameron, Seth R Flaxman, Daniel J Weiss, David L Smith, and Peter W Gething. Improved prediction accuracy for disease risk mapping using gaussian process stacked generalization. *Journal of The Royal Society Interface*, 14(134):20170520, 2017.

[15] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[16] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *arXiv preprint arXiv:1712.03141*, 2017.

[17] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[18] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[19] Arno Blaas, Luca Laurenti, Andrea Patane, Luca Cardelli, Marta Kwiatkowska, and Stephen Roberts. Robustness quantification for classification with gaussian processes. *arXiv preprint arXiv:1905.11876*, 2019.

[20] Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 5760–5770, 2018.

[21] Luca Bortolussi, Luca Cardelli, Marta Kwiatkowska, and Luca Laurenti. Central limit model checking. *arXiv preprint arXiv:1804.08744*, 2018.

[22] Tibor Bosse, Matthijs Pontier, and Jan Treur. A computational model based on gross' emotion regulation theory. *Cognitive systems research*, 11(3):211–230, 2010.

[23] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[24] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

[25] Michael Brennan, Marimuthu Palaniswami, and Peter Kamen. Do existing measures of poincare plot geometry reflect nonlinear features of heart rate variability? *IEEE transactions on biomedical engineering*, 48(11):1342–1347, 2001.

[26] Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. Robustness of bayesian neural networks to gradient-based attacks. *In Advances in neural information processing systems (to appear)*, 2020.

[27] L Cardelli, M Kwiatkowska, L Laurenti, N Paoletti, A Patane, and M Wicker. Statistical guarantees for the robustness of bayesian neural networks. International Joint Conference on Artificial Intelligence, 2019.

[28] Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, and Andrea Patane. Robustness guarantees for bayesian inference with gaussian processes. 33:7759–7768, 2019.

[29] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[30] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[31] Sih-Huei Chen, Yuan-Shan Lee, Wen-Chi Hsieh, and Jia-Ching Wang. Music emotion recognition using deep gaussian process. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 495–498. IEEE, 2015.

[32] Fei Cheng, Jiangsheng Yu, and Huilin Xiong. Facial expression recognition in jaffe dataset based on gaussian process classification. *IEEE Transactions on Neural Networks*, 21(10):1685–1690, 2010.

[33] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[34] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.

[35] et al Clifton, Lei. Gaussian process regression in vital-sign early warning systems. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6161–6164. IEEE, 2012.

[36] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[37] Desirée Colombo, Javier Fernández-Álvarez, Andrea Patané, Michelle Semonella, Marta Kwiatkowska, Azucena García-Palacios, Pietro Cipresso, Giuseppe Riva, and Cristina Botella. Current state and future directions of technology-based ecological momentary assessment and intervention for major depressive disorder: A systematic review. *Journal of clinical medicine*, 8(4):465, 2019.

[38] Desirée Colombo, Azucena Garcia Palacios, Javier Fernandez Alvarez, Andrea Patané, Michelle Semonella, Pietro Cipresso, Marta Kwiatkowska, Giuseppe Riva, and Cristina Botella. Current state and future directions of technology-based ecological momentary assessments and interventions for major depressive disorder: protocol for a systematic review. *Systematic reviews*, 7(1):233, 2018.

[39] Vanessa P da Silva, Bruno Ribeiro Ramalho Oliveira, Tavares Mello, Roger Gomes, Helena Moraes, Andrea Camaz Deslandes, and Jerson Laks. Heart rate variability indexes in dementia: A systematic review with a quantitative analysis. *Current Alzheimer Research*, 15(1):80–88, 2018.

[40] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

[41] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[42] Sina Däubener, Lea Schönherr, Asja Fischer, and Dorothea Kolossa. Detecting adversarial examples for speech recognition via uncertainty quantification. *arXiv preprint arXiv:2005.14611*, 2020.

[43] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[44] Rahul C Deo. Machine learning in medicine. *Circulation*, 132(20):1920–1930, 2015.

[45] Sidney K D'mello and Jacqueline Kory. A review and meta-analysis of multimodal affect detection systems. *ACM Computing Surveys (CSUR)*, 47(3):1–36, 2015.

[46] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[47] Richard M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

[48] Robert Dürichen, Marco AF Pimentel, Lei Clifton, Achim Schweikard, and David A Clifton. Multitask gaussian processes for multivariate physiological time-series analysis. *IEEE Transactions on Biomedical Engineering*, 62(1):314–322, 2014.

[49] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[50] Krishnamurthy Dvijotham, Marta Garnelo, Alhussein Fawzi, and Pushmeet Kohli. Verification of deep probabilistic models. *arXiv preprint arXiv:1812.02795*, 2018.

[51] Simon Eberz, Giulio Lovisotto, Andrea Patane, Marta Kwiatkowska, Vincent Lenders, and Ivan Martinovic. When your fitness tracker betrays you: Quantifying the predictability of biometric features across contexts. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 889–905. IEEE, 2018.

[52] Gernot Ernst. *Heart rate variability*. Springer, 2016.

[53] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[54] Debin Fang, Xiaoling Zhang, Qian Yu, Trenton Chen Jin, and Luan Tian. A novel method for carbon dioxide emission forecasting based on improved gaussian processes regression. *Journal of cleaner production*, 173:143–150, 2018.

[55] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

[56] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[57] C Fere. Note on changes in electrical resistance under the effect of sensory stimulation and emotion. *Comptes rendus des seances de la societé de biologie*, 5:217–219, 1888.

[58] Antonio Fernández-Caballero, Arturo Martínez-Rodrigo, José Manuel Pastor, José Carlos Castillo, Elena Lozano-Monasor, María T López, Roberto Zangróniz, José Miguel Latorre, and Alicia Fernández-Sotos. Smart environment architecture for emotion detection and regulation. *Journal of Biomedical Informatics*, 64:55–73, 2016.

[59] Samuel G Finlayson, Hyung Won Chung, Isaac S Kohane, and Andrew L Beam. Adversarial attacks against medical deep learning systems. *arXiv preprint arXiv:1804.05296*, 2018.

[60] Pooria Sarrami Foroushani, Justine Schneider, and Neda Assareh. Meta-review of the effectiveness of computerised cbt in treating depression. *BMC psychiatry*, 11(1):131, 2011.

[61] David T Frazier and Christopher Drovandi. Robust approximate bayesian inference with synthetic likelihood. *arXiv preprint arXiv:1904.04551*, 2019.

[62] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[63] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[64] Yarin Gal and Lewis Smith. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with Bayesian neural networks. *arXiv preprint arXiv:1806.00667*, 2018.

[65] Pei Gao, Antti Honkela, Magnus Rattray, and Neil D Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24(16):i70–i75, 2008.

[66] Hernán F García, Mauricio A Álvarez, and Álvaro A Orozco. Gaussian process dynamical models for multimodal affect recognition. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 850–853. IEEE, 2016.

[67] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[68] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*, 2018.

[69] Patrick Gebhard. Alma: a layered model of affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36. ACM, 2005.

[70] S Ghiasi, A Patane, A Greco, L Laurenti, EP Scilingo, and M Kwiatkowska. Gaussian processes with physiologically-inspired priors for physical arousal recognition.

[71] Shadi Ghiasi, Alberto Greco, Riccardo Barbieri, Enzo Pasquale Scilingo, and Gaetano Valenza. Assessing autonomic function from electrodermal activity and heart rate variability during cold-pressor test and emotional challenge. *Scientific Reports*, 10(1):1–13, 2020.

[72] Shadi Ghiasi, Alberto Greco, Mimma Nardelli, Vincenzo Catrambone, Riccardo Barbieri, Enzo Pasquale Scilingo, and Gaetano Valenza. Investigating phasic activity of time-varying high-order spectra: A heartbeat dynamics study during cold-pressor test. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[73] Shadi Ghiasi, Andrea Patane, Luca Laurenti, Enzo Pasquale Scilingo, Alberto Greco, and Marta Kwiatkowska. Physiologically informed gaussian processes for modelling of psycho-physiological states. *Under review*, 2020.

[74] John Gideon, Soheil Khorram, Zakaria Aldeneh, Dimitrios Dimitriadis, and Emily Mower Provost. Progressive neural networks for transfer learning in emotion recognition. *arXiv preprint arXiv:1706.03256*, 2017.

[75] Marco Gomes, Tiago Oliveira, Fábio Silva, Davide Carneiro, and Paulo Novais. Establishing the relationship between personality traits and stress in an intelligent environment. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 378–387. Springer, 2014.

[76] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[77] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[78] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[79] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[80] Francesco Massa Gray and Michael Schmidt. Thermal building modelling using gaussian processes. *Energy and Buildings*, 119:119–128, 2016.

[81] Alberto Greco, Gaetano Valenza, Antonio Lanata, Enzo Pasquale Scilingo, and Luca Citi. cvxeda: A convex optimization approach to electrodermal activity processing. *IEEE Transactions on Biomedical Engineering*, 63(4):797–804, 2016.

[82] Alberto Greco, Gaetano Valenza, and Enzo Pasquale Scilingo. *Advances in Electrodermal activity processing with applications for mental health*. Springer, 2016.

[83] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[84] Kathrin Grosse, David Pfaff, Michael T Smith, and Michael Backes. The limitations of model uncertainty in adversarial settings. *arXiv preprint arXiv:1812.02606*, 2018.

[85] Kathrin Grosse, David Pfaff, Michael Thomas Smith, and Michael Backes. How wrong am i?-studying adversarial examples and their impact on uncertainty in gaussian process machine learning models. *arXiv preprint arXiv:1711.06598*, 2017.

[86] Muhammad Abdullah Hanif, Faiq Khalid, Rachmad Vidya Wicaksana Putra, Semeen Rehman, and Muhammad Shafique. Robust machine learning systems: Reliability and security for deep neural networks. In *2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pages 257–260. IEEE, 2018.

[87] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65, 2019.

[88] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[89] Jennifer A Healey and Rosalind W Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on intelligent transportation systems*, 6(2):156–166, 2005.

[90] Daniel Hernández-Lobato, José M Hernández-Lobato, and Pierre Dupont. Robust multi-class Gaussian process classification. In *Advances in neural information processing systems*, pages 280–288, 2011.

[91] Kristin E Heron and Joshua M Smyth. Ecological momentary interventions: incorporating mobile technology into psychosocial and health behaviour treatments. *British journal of health psychology*, 15(1):1–39, 2010.

[92] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, Mingyue Niu, and Minghao Yang. Multimodal continuous emotion recognition with data augmentation using recurrent neural networks. In *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, pages 57–64, 2018.

[93] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

[94] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.

[95] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. In *Advances in Neural Information Processing Systems*, pages 15883–15893, 2019.

[96] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[97] Syed Muhammad Imaduddin, Andrea Fanelli, Frederick Vonberg, Robert C Tasker, and Thomas Heldt. Pseudo-bayesian model-based noninvasive intracranial pressure estimation and tracking. *IEEE Transactions on Biomedical Engineering*, 2019.

[98] John Jackson, Luca Laurenti, Eric Frew, and Morteza Lahijanian. Safety verification of unknown dynamical systems via gaussian process regression. *arXiv preprint arXiv:2004.01821*, 2020.

[99] Natasha Jaques, Sara Taylor, Akane Sano, Rosalind Picard, et al. Predicting tomorrow's mood, health, and stress level using personalized multitask learning and domain adaptation. In *IJCAI 2017 Workshop on artificial intelligence in affective computing*, pages 17–33, 2017.

[100] Jesper Jeppesen, Anders Fuglsang-Frederiksen, Ramon Brugada, Birthe Pedersen, Guido Rubboli, Peter Johansen, and Sándor Beniczky. Heart rate variability analysis indicates preictal parasympathetic overdrive preceding seizure-induced cardiac dysrhythmias leading to sudden unexpected death in a patient with epilepsy. *Epilepsia*, 55(7), 2014.

[101] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 259–274, 2019.

[102] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 864–872, 2019.

[103] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[104] Kyle D Julian and Mykel J Kochenderfer. Guaranteeing safety for neural network-based aircraft collision avoidance systems. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2019.

[105] Markus Kächele, Patrick Thiam, Mohammadreza Amirian, Philipp Werner, Steffen Walter, Friedhelm Schwenker, and Günther Palm. Multimodal data

fusion for person-independent, continuous estimation of pain intensity. In *International Conference on Engineering Applications of Neural Networks*, pages 275–285. Springer, 2015.

[106] Kazuya Kakizaki, Kosuke Yoshida, and Tsubasa Takahashi. Glassmasq: Adversarial examples masquerading in face identification systems with feature extractor. In *2019 17th International Conference on Privacy, Security and Trust (PST)*, pages 1–7. IEEE, 2019.

[107] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[108] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[109] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian Gaussian process classification with the EM-EP algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1948–1959, 2006.

[110] Hyun-Chul Kim and Zoubin Ghahramani. Outlier robust Gaussian process classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 896–905. Springer, 2008.

[111] Kyung Hwan Kim, Seok Won Bang, and Sang Ryong Kim. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, 42(3):419–427, 2004.

[112] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust bayesian optimization. *arXiv preprint arXiv:2002.09038*, 2020.

[113] William Kleiber, Richard W Katz, and Balaji Rajagopalan. Daily spatiotemporal precipitation simulation using latent and transformed gaussian processes. *Water Resources Research*, 48(1), 2012.

[114] Rafal Kocielnik, Natalia Sidorova, Fabrizio Maria Maggi, Martin Ouwerkerk, and Joyce HDM Westerink. Smart technologies for long-term stress monitoring at work. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pages 53–58. IEEE, 2013.

[115] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

[116] Tomoki Koriyama and Takao Kobayashi. Semi-supervised prosody modeling using deep gaussian process latent variable model. In *INTERSPEECH*, pages 4450–4454, 2019.

[117] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[118] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[119] Luca Laurenti, Alessandro Abate, Luca Bortolussi, Luca Cardelli, Milan Ceska, and Marta Kwiatkowska. Reachability computation for switching diffusions: Finite abstractions with certifiable and tuneable precision. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 55–64. ACM, 2017.

[120] Luca Laurenti, Morteza Lahijanian, Alessandro Abate, Luca Cardelli, and Marta Kwiatkowska. Formal and efficient synthesis for continuous-time linear stochastic hybrid processes. *IEEE Transactions on Automatic Control*, 2020.

[121] Luca Laurenti, Andrea Patane, Matthew Wicker, Luca Bortolussi, Luca Cardelli, and Marta Kwiatkowska. Global adversarial robustness guarantees for neural networks. 2019.

[122] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[123] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[124] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. *arXiv preprint arXiv:1703.02914*, 2017.

[125] Changliu Liu, Tomer Arnon, Christopher Lazarus, Clark Barrett, and Mykel J Kochenderfer. Algorithms for verifying deep neural networks. *arXiv preprint arXiv:1903.06758*, 2019.

[126] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *arXiv preprint arXiv:1810.01279*, 2018.

[127] Zhen Liu and Zhi Geng Pan. An emotion model of 3d virtual characters in intelligent virtual environment. In *International Conference on Affective Computing and Intelligent Interaction*, pages 629–636. Springer, 2005.

[128] Daniel Lopez-Martinez and Rosalind Picard. Multi-task neural networks for personalized pain recognition from physiological signals. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 181–184. IEEE, 2017.

[129] Daniel Lopez-Martinez and Rosalind Picard. Continuous pain intensity estimation from autonomic signals with recurrent neural networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5624–5627. IEEE, 2018.

[130] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[131] David JC MacKay. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992.

[132] Stacy Marsella, Jonathan Gratch, Paolo Petta, et al. Computational models of emotion. *A Blueprint for Affective Computing-A sourcebook and manual*, 11(1):21–46, 2010.

[133] Hector P Martinez, Yoshua Bengio, and Georgios N Yannakakis. Learning deep physiological models of affect. *IEEE Computational intelligence magazine*, 8(2):20–33, 2013.

[134] Javier Mateo and Pablo Laguna. Improved heart rate variability signal analysis from the beat occurrence times according to the ipfm model. *IEEE Transactions on Biomedical Engineering*, 47(8):985–996, 2000.

[135] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[136] Alexander Graeme de Garis Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2017.

[137] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, Ro. Cipolla, and A. Weller. Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. IJCAI, 2017.

[138] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[139] Paolo Melillo, Marcello Bracale, and Leandro Pecchia. Nonlinear heart rate variability features for real-life stress detection. case study: students under stress due to university examination. *Biomedical engineering online*, 10(1):96, 2011.

[140] Paolo Melillo, Raffaele Izzo, Ada Orrico, Paolo Scala, Marcella Attanasio, Marco Mirra, Nicola De Luca, and Leandro Pecchia. Automatic prediction of cardiovascular and cerebrovascular events using heart rate variability analysis. *PloS one*, 10(3):e0118504, 2015.

[141] Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. *arXiv preprint arXiv:1909.09884*, 2019.

[142] Raúl Quintero Mínguez, Ignacio Parra Alonso, David Fernández-Llorca, and Miguel Ángel Sotelo. Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1803–1814, 2018.

[143] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[144] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[145] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[146] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[147] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[148] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13:271–369, 2004.

[149] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in neural information processing systems*, pages 5947–5956, 2017.

[150] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

[151] Simon Ollander, Christelle Godin, Sylvie Charbonnier, and Aurélie Campagne. Feature and sensor selection for detection of driver stress. In *PhyCS*, pages 115–122, 2016.

[152] Nicola Paoletti, Andrea Patanè, and Marta Kwiatkowska. Closed-loop quantitative verification of rate-adaptive pacemakers. *ACM Transactions on Cyber-Physical Systems*, 2(4):33, 2018.

[153] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[154] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[155] Hyunsin Park and Chang D Yoo. Gaussian process dynamical models for phoneme classification. In *NIPS 2011 Workshop on Bayesian Nonparametrics: Hope or Hype*. Citeseer, 2011.

[156] Andrea Patane, Shadi Ghiasi, Enzo Pasquale Scilingo, and Marta Kwiatkowska. Automated recognition of sleep arousal using multimodal and personalized deep ensembles of neural networks. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[157] Andrea Patanè and Marta Kwiatkowska. Calibrating the classifier: Siamese neural network architecture for end-to-end arousal recognition from ecg. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 1–13. Springer, 2018.

[158] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[159] Rosalind W Picard, Szymon Fedor, and Yadid Ayzenberg. Multiple arousal theory and daily-life electrodermal activity asymmetry. *Emotion Review*, 8(1):62–75, 2016.

[160] Rosalind W. Picard, Elias Vyzas, and Jennifer Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE transactions on pattern analysis and machine intelligence*, 23(10):1175–1191, 2001.

[161] Rosalind Wright Picard et al. Affective computing. 1995.

[162] et al. Pimentel, Marco AF. Probabilistic estimation of respiratory rate using gaussian processes. In *2013 35th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2902–2905. IEEE, 2013.

[163] Kyriakos Polymenakos, Luca Laurenti, Andrea Patane, Jan-Peter Calliess, Luca Cardelli, Marta Kwiatkowska, Alessandro Abate, and Stephen Roberts. Safety guarantees for planning based on iterative gaussian processes. *in Conference on Decision and Control (to appear)*, 2020.

[164] Adnan Qayyum, Junaid Qadir, Muhammad Bilal, and Ala Al-Fuqaha. Secure and robust machine learning for healthcare: A survey. *arXiv preprint arXiv:2001.08103*, 2020.

[165] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International Conference on Machine Learning*, pages 5231–5240. PMLR, 2019.

[166] Qing Rao and Jelena Frtunikj. Deep learning for self-driving cars: chances and challenges. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, pages 35–38, 2018.

[167] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[168] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of machine learning research*, 11(Nov):3011–3015, 2010.

[169] Ambrish Rawat, Martin Wistuba, and Maria-Irina Nicolae. Adversarial phenomenon in the eyes of bayesian deep learning. *arXiv preprint arXiv:1711.08244*, 2017.

[170] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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. ACM, 2016.

[171] Stefano Rosa, Andrea Patane, Chris Xiaoxuan Lu, and Niki Trigoni. Semantic place understanding for human–robot coexistence—toward intelligent workplaces. *IEEE Transactions on Human-Machine Systems*, 49(2):160–170, 2018.

[172] Stefano Rosa, Andrea Patanè, Xiaoxuan Lu, and Niki Trigoni. Commonsense: Collaborative learning of scene semantics by robots and humans. In *Proceedings of the 1st International Workshop on Internet of People, Assistive Robots and Things*, pages 1–6. ACM, 2018.

[173] J Ben Rosen and Panos M Pardalos. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34(2):163–174, 1986.

[174] Philipp V Rouast, Marc Adam, and Raymond Chiong. Deep learning for human affect recognition: insights and new developments. *IEEE Transactions on Affective Computing*, 2019.

[175] Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. *arXiv preprint arXiv:1805.02242*, 2018.

[176] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[177] Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 927–934. Citeseer, 2010.

[178] Dorsa Sadigh and Ashish Kapoor. Safe control under uncertainty. *arXiv preprint arXiv:1510.07313*, 2015.

[179] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

[180] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[181] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. A review of emotion recognition using physiological signals. *Sensors*, 18(7):2074, 2018.

[182] Siddharth Siddharth, Tzyy-Ping Jung, and Terrence J Sejnowski. Utilizing deep learning towards multi-modal bio-sensing and vision-based affective computing. *IEEE Transactions on Affective Computing*, 2019.

[183] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018.

[184] Michael Thomas Smith, Kathrin Grosse, Michael Backes, and Mauricio A Alvarez. Adversarial vulnerability bounds for gaussian process classification. *arXiv preprint arXiv:1909.08864*, 2019.

[185] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

[186] Oliver Stegle, Sebastian V Fallert, David JC MacKay, and Søren Brage. Gaussian process robust regression for noisy heart rate data. *IEEE Transactions on Biomedical Engineering*, 55(9):2143–2151, 2008.

[187] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[188] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International Conference on Machine Learning*, pages 997–1005, 2015.

[189] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[190] Sara Ann Taylor, Natasha Jaques, Ehimwenma Nosakhare, Akane Sano, and Rosalind Picard. Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*, 2017.

[191] Patrick Thiam, Peter Bellmann, Hans A Kestler, and Friedhelm Schwenker. Exploring deep physiological models for nociceptive pain recognition. *Sensors*, 19(20):4503, 2019.

[192] Richard Tomsett, Amy Widdicombe, Tianwei Xing, Supriyo Chakraborty, Simon Julier, Prudhvi Gurram, Raghuveer Rao, and Mani Srivastava. Why the failure? how adversarial examples can provide insights for interpretable machine learning. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 838–845. IEEE, 2018.

[193] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[194] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[195] Raquel Urtasun, David J Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 238–245. IEEE, 2006.

[196] Gaetano Valenza, Luca Citi, Enzo Pasquale Scilingo, and Riccardo Barbieri. Point-process nonlinear models with laguerre and volterra expansions: Instantaneous assessment of heartbeat dynamics. *IEEE Transactions on Signal Processing*, 61(11):2914–2926, 2013.

[197] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14(Apr):1175–1179, 2013.

[198] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[199] Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. Safe exploration and optimization of constrained mdps using gaussian processes. In *AAAI Conference on Artificial Intelligence*, 2018.

[200] Steffen Walter, Sascha Gruss, Hagen Ehleiter, Junwen Tan, Harald C Traue, Philipp Werner, Ayoub Al-Hamadi, Stephen Crawcour, Adriano O Andrade, and Gustavo Moreira da Silva. The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. In *2013 IEEE international conference on cybernetics (CYBCO)*, pages 128–131. IEEE, 2013.

[201] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019.

[202] Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. Statistical verification of neural networks. *arXiv preprint arXiv:1811.07209*, 2018.

[203] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

[204] Philipp Werner, Ayoub Al-Hamadi, Robert Niese, Steffen Walter, Sascha Gruss, and Harald C Traue. Automatic pain recognition from video and biomedical signals. In *2014 22nd International Conference on Pattern Recognition*, pages 4582–4587. IEEE, 2014.

[205] Philipp Werner, Daniel Lopez-Martinez, Steffen Walter, Ayoub Al-Hamadi, Sascha Gruss, and Rosalind Picard. Automatic recognition methods supporting pain assessment: A survey. *IEEE Transactions on Affective Computing*, 2019.

[206] Matthew Wicker, Xiaowei Huang, and Marta Kwiatkowska. Feature-guided black-box safety testing of deep neural networks. In *International Conferfence on Tools and Algorithms for the Construction and Analysis of Systems*, pages 408–426. Springer, 2018.

[207] Matthew Wicker, Luca Laurenti, Andrea Patane, and Marta Kwiatkowska. Probabilistic safety for bayesian neural networks. *In Uncertainty in Artificial Intelligence (to appear)*, 2020.

[208] Christopher KI Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[209] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[210] Andrew Gordon Wilson. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, Citeseer, 2014.

[211] Josef Wilzen, Anders Eklund, and Mattias Villani. Physiological gaussian process priors for the hemodynamics in fmri analysis. *arXiv preprint arXiv:1708.06152*, 2017.

[212] Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. Adversarial examples in modern machine learning: A review. *arXiv preprint arXiv:1911.05268*, 2019.

[213] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, pages 8400–8409, 2018.

[214] Okito Yamashita, Andreas Galka, Tohru Ozaki, Rolando Biscay, and Pedro Valdes-Sosa. Recursive penalized least squares solution for dynamical inverse problems of eeg generation. *Human brain mapping*, 21(4):221–235, 2004.

[215] Ke Yan and David Zhang. Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sensors and Actuators B: Chemical*, 212:353–363, 2015.

[216] Greg Yang. Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *arXiv preprint arXiv:1910.12478*, 2019.

[217] Nanyang Ye and Zhanxing Zhu. Bayesian adversarial learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6892–6901, 2018.

[218] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI conference on artificial intelligence*, 2017.

[219] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[220] Wen Zhu, Nancy Zeng, Ning Wang, et al. Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, 19:67, 2010.

[221] Roland S Zimmermann. Comment on" adv-bnn: Improved adversarial defense through robust bayesian neural network". *arXiv preprint arXiv:1907.00895*, 2019.