Quantifying Uncertainty in Deep Learning

Quantifying Uncertainty in Deep Learning

Taras Holovko

A thesis submitted to the Committee
on Degrees in Applied Mathematics
in partial fulfillment of the
requirements for the degree of

Bachelor of Arts

Harvard College
April 2020

Acknowledgements

This thesis would not be possible without the support of many people.

I would first and foremost like to thank my advisor, Weiwei Pan, who inspired my interest in Bayesian deep learning as my professor in a graduate applied mathematics course on stochastic methods. I am so grateful for her guidance, patience, and understanding throughout this entire process.

I would also like to thank Sujay Thakur and Yaniv Yacoby from my research group for their helpful comments and assistance with code.

Finally, I would like to thank my parents for supporting me in every step of my education up until this point, which has culminated in this thesis.

Abstract

Deep learning literature has witnessed an abundance of proposals for novel models of uncertainty in recent years. However, there has been comparatively little emphasis on the need for separate estimates for aleatoric and epistemic uncertainty, which are uniquely different types of uncertainties that arise from different sources and, as such, have different implications for real-life decision-making, especially in safety-critical contexts such as medical diagnosis.

In this thesis, we contribute to the literature a systematic and comparative evaluation of different metrics for quantifying aleatoric and epistemic uncertainty. In particular, we consider estimates for uncertainty that arise from traditional measures of variability in categorical distributions, decompositions of total uncertainty that are based upon classical statistical principles, and out-of-distribution detection metrics. We pair evaluation of all of these metrics with a variety of different models and inference methods that are rooted in both traditional and Bayesian deep learning.

We extend two separate decompositions of aleatoric and epistemic uncertainty to deep ensembles and statistical measures of variability in novel ways, and we evaluate both approaches as providing accurate estimates. Finally, we evaluate Monte Carlo dropout as an inference method applied to both homoscedastic and heteroscedastic regression models, and we find that it does not produce accurate aleatoric and epistemic uncertainty estimates as is suggested in the literature.

Table of contents

Page

Appendix

Chapter One

Introduction

At its core, machine learning is a process of inductive learning that seeks to generalize beyond data. Per this process, models are constructed based upon observed data and attempt to make predictions that accurately reflect the true mechanisms underlying the domain from which data is sampled or, in statistical terms, the underlying data-generating process. However, because the true data-generating processes are not known with certainty in the domains where machine learning methods are applied—in real-life applications, at the least—specific models cannot be proven correct. By extension, these models must be inherently uncertain, and so must their predictions.

While there have been great advances in the accuracy of machine learning models in recent years, less emphasis has been placed historically on accurately representing uncertainty within such models. In particular, one subset of methods that have achieved renown for their accuracy as universal approximators (Cybenko, 1989) is the broad category of *deep learning* models, in which collections of connected computational units known as *artificial neural networks* learn to perform tasks through an iterative updating process of the parameters of these units. Literature on deep learning often reports models achieving high predictive accuracy across many domains. At the same time, however, there have been fewer strides towards quantifying and understanding predictive uncertainties—which may be rooted in model assumptions or noisy (or, at times, nonexistent) data—accompanied with the individual predictions of such models. Given that deep learning is beginning to be deployed in safety-critical applications—disease diagnosis, autonomous driving to name a few—there is a clear need for predictive uncertainty estimates that are at once accurate and informative.

Probabilistic modeling is the conventional way to represent uncertainty, but traditional deep learning models do not adopt this approach. Rather, traditional deep learning models are trained to minimize some type of *loss* on training data, which does not necessarily have probabilistic interpretations. These models do not inherently take uncertainty into account; instead, model training yields point estimates of parameters and outputs, ignoring the uncertainty over different model possibilities. Consequently, such models tend to provide overconfident predictions, particularly in regions where data is scarce or unavailable.

In light of this, an alternative to traditional deep learning that is rooted in probabilistic modelling—a family of methods known as Bayesian deep learning (BDL)—has gained popularity in recent years. In contrast to traditional deep learning, BDL methods use probability distributions to model the weights of a neural network. This equips such models to represent uncertainty by providing probabilistic estimates for both model parameters and target variable predictions. Moreover, BDL approaches have become increasingly accessible as advances in Monte Carlo sampling and variational inference (VI), which are required practically to explore their intricate posterior distributions, have improved their viability for application to real-life tasks.

However, insofar as Bayesian approaches offer a way to represent uncertainty probabilistically, one question to which even BDL has not offered a clear solution is the decomposition and quantification of two inherently different types of uncertainty, referred to as *aleatoric* and *epistemic* uncertainty. *Aleatoric* uncertainty, often referenced as statistical uncertainty, refers to an inherent notion of randomness or variability in an experiment or process that will always exist. On the other hand, *epistemic* uncertainty, also referenced as systemic uncertainty, is uncertainty that exists due to a lack of knowledge and, by implication, that can be eliminated with additional knowledge or information (Hüllermeier and Waegeman, 2019). From a functional viewpoint, we think about these two uncertainties as referring to the irreducible and reducible components, respectively, of total uncertainty.

An illustrative example that demonstrates both types of uncertainty is the roll of a biased die, a situation in which some numbers are more likely to be rolled than others but wherein the die roller—or decision maker—is unaware of the exact bias of the die. In this scenario, there is an element of uncertainty that exists due to the decision maker's lack of knowledge about the exact bias of the die. This component of uncertainty, the epistemic, can be reduced if the decision maker obtains additional information about the die's bias through the collection of additional data; knowing that it is loaded to favor a particular side, for instance, reduces the decision maker's uncertainty about outcomes in the long term. By comparison, there is an element of stochastic uncertainty in the roll of the die that will always exist irrespective of the decision maker's epistemic state. Even if the decision maker were perfectly informed about the exact bias of the die, there is inherent randomness in what the next roll will yield; this is the irreducible, or aleatoric, component of uncertainty.

While these two uncertainties are typically not distinguished in deep learning, we argue in the following sections of the introduction that there are clear and important stakes for doing so. We first provide philosophical background and context for how aleatoric and epistemic uncertainty are defined across literature in §1.1. Then, having established these definitions, we argue for the importance of distinguishing these two uncertainties in real-life decision-making contexts in §1.2. With these stakes in mind in §1.3, we outline how aleatoric and epistemic uncertainty are quantified in deep learning, noting that many of the approaches cannot be applied in a model-agnostic, adaptable way. Finally, in §1.4, we will describe how this thesis will seek to fill gaps in the existing machine learning literature and will outline the structure of the thesis.

In short, this thesis sets out to evaluate and compare flexible and model-agnostic metrics for representing and quantifying aleatoric and epistemic uncertainty. These metrics are drawn from several domains and applied to multiple synthetic and real downstream tasks.

1.1 Defining aleatoric and epistemic uncertainty

Literature concerning aleatoric and epistemic uncertainty outside of the realm of machine learning stems from numerous fields, including statistics, philosophy, engineering, and structural safety. Although works across disciplines define aleatoric and epistemic uncertainty with slight differences, the underlying common ground is that epistemic uncertainty is attributed to the state of knowledge of the agent and can (and perhaps should) be reduced, whereas aleatoric uncertainty cannot be reduced and therefore should be managed in some way.

While many frameworks have been proposed for categorizing uncertainty in literature, there are two particularly instructive concepts that contextualize the difficulties of using classical statistical methods—and probabilistic tools more broadly—to represent epistemic and aleatoric uncertainty. The first pertains to the distinction between the class-based and continuous nature of aleatoric uncertainty and the case-based and binary nature of epistemic uncertainty (this also highlights the divergence between frequentist and Bayesian perspectives). The second pertains to the notion of uncertainties as being rooted internally or externally and to the ways in which decision makers typically handle them, and as such implies that there are disadvantages to using probability theory to model uncertainty altogether.

1.1.1 Case-based versus class-based uncertainty

C. R. Fox and Ülkümen (2011), in reviewing judgment and decision making literature to identify the characteristics of epistemic and aleatoric uncertainty, describes epistemic uncertainty as case-based and aleatoric uncertainty as class-based, a difference that is closely intertwined with the philosophical split between the frequentist and Bayesian schools of thought in statistics.

Per C. R. Fox and Ülkümen (2011), epistemic uncertainty corresponds to the evaluation of *single* events in terms of *binary* truth value. One's state of epistemic certainty regarding whether an event is true or false is attributed to insufficient knowledge; uncertainty can be reduced if the individual searches for further information, patterns, or causality. Moreover, from a psychological standpoint, an individual's conception of the epistemic uncertainty of an event is linguistically linked to the confidence that the individual has regarding the outcome of the event. This is so much so, in fact, that judgments of events that are purely epistemic are disproportionately influenced by differences in evidence strength and, as such, tend toward probabilities of 0 or 1 more often than that of aleatoric events.

By comparison, evaluating aleatoric uncertainty usually entails considering a *class* of possible outcomes and evaluation of the propensity of each event on a *continuous* scale. Individuals conceptualize the aleatoric uncertainty of events in terms of relative frequency of outcomes and associate it linguistically with probability and chance. Aleatoric judgment, in neurobiological terms, entails a cognitive process distinct from the one associated with epistemic judgment (Volz, Schubotz, and Cramon, 2005).

These distinct patterns of reasoning associated with aleatoric and epistemic uncertainty have also been described as corresponding to *frequentist* and *nonfrequentist* events, which are conceived to be uniquely different (Howell and Burnett, 1978) and identified in closely related terms in other literature (Gigerenzer, 1994; Peterson and Pitz, 1988).

Frequentist versus Bayesian perspectives

The categorization of frequentist and nonfrequentist events reflects the difficulty of expressing epistemic uncertainty with an approach rooted in classical—also known as *frequentist*—statistics.

The frequentist school of thought believes that probabilities represent the long-run frequencies with which events occur and thus can be found through a repeatable, objective process and represented without opinion. By implication, because epistemic uncertainty is usually framed in terms of the occurrence of single, non-repeating events as being true or false, it cannot be expressed in frequentist terms and therefore cannot be quantified as a probability. In this vein, frequentist inference methods such as confidence intervals or significance tests, which are commonly misinterpreted as making probability statements about parameters, only describe potential outcomes in terms of repeated sampling. Indeed, they do not state anything directly regarding the single, non-repeating event of interest.

By comparison, Bayesian thinking offers a perspective that—unlike the frequentist one—allows us to express non-repeating events in probabilistic terms. In the Bayesian school of thought, probabilities represent a degree of belief in the truth of a proposition and are therefore considered to be subjective. Inference tests in Bayesian statistics describe how the acquisition of new information, i.e., data, reduces uncertainty about such a proposition

(O'Hagan, 2004). The process of using data to modify a *prior* probability, which represents an initial belief, to compute a *posterior* probability, which represents an updated belief, is known as *Bayesian updating*.

Whereas frequentist inference cannot make direct statements about events of an epistemic nature because it relies on repeated sampling, Bayesian inference makes statements that are unambiguously about these very events (O'Hagan, 2004). Moreover, the Bayesian degree-of-belief interpretation of probability naturally lends itself to expressing the binary truth values that characterize epistemic uncertainty.

One instructive example that demonstrates the advantage of employing a Bayesian rather than frequentist approach concerns estimating parameters for deep learning models. Because model parameters are almost always considered to be epistemically uncertain, frequentist statistics is unequipped to convey this uncertainty and instead expresses parameters with point estimates, which are found using *maximum likelihood estimation* (MLE). Because of this, traditional deep learning models that are grounded in frequentist thought, as previously mentioned, are uninformative with respect to epistemic uncertainty.

One solution to this problem that is considered to be frequentist has been proposed in the form of *ensemble* methods, which use the outputs of multiple deep learning models to improve predictive performance and also create distributions over estimates (Lakshminarayanan, Pritzel, and Blundell, 2017). Unlike BDL, ensembles can be implemented without any reliance on priors. For this reason, one might claim that ensembles are more empirical, allowing for—as frequentists might say—the data to speak for itself. Moreover, ensembles are more computationally practical than expensive BDL models, which require the approximation of intractable posterior landscapes.

On the other hand, however, ensemble methods—while advantageous in some respects—are not principled in the same way that BDL methods are. This is to say, whereas BDL directly conveys any assumptions regarding the selection of priors, frequentists do not expose the model assumptions that may intrinsically inform how the data interacts with the model. For instance, the selection of specific activation functions—for both BDL and ensembles—alters how the model represents variance in the data, but an ensemble approach can hypothetically allow for the model to demonstrate infinite uncertainty, whereas BDL limits this possibility, restricting the possible range of uncertainty with the selection of particular priors. Resultantly, the diversity of model outputs in ensembles tends to be more dependent on the model class and assumptions, at least in a sense that cannot be expressed theoretically through the selection of a prior.

We directly compare Bayesian and ensemble approaches in their ability to quantify uncertainty, as captured with a variety of metrics, in the second half of this work.

1.1.2 Internal versus external uncertainty and probability theory

Although we have established in the preceding section that probability distributions—whether constructed by way of Bayesian or ensemble approaches—are commonly used to represent uncertainty, scholars have questioned whether probability theory is well-equipped to represent epistemic uncertainty altogether. In fact, the argument that a single probability distribution is insufficient for representing ignorance is commonly maintained in the literature (Hüllermeier and Waegeman, 2019). This line of reasoning, which maintains that probabilistic tools do not inherently differentiate between awareness of statistical randomness and lack of knowledge, can be attributed philosophically to the distinction between *internal* and *external* uncertainty and, more fundamentally, to the agency that decision makers have to reduce each of these uncertainties.

Broadly speaking, internal uncertainty corresponds to one's state of knowledge, whereas external uncertainty is attributed to the external world and is further divided into *singular* and *distributional* modes. The singular mode refers to scenarios in which probabilities are evaluated in terms of the propensity of a particular target event, whereas the distributional mode refers to scenarios in which the event at hand is thought to be an instance of a class of similar events (Kahneman and Tversky, 1982). C. R. Fox and Ülkümen (2011) maintains that the external-distributional mode generally maps to aleatoric uncertainty and the internal and external-singular modes map to epistemic uncertainty.

This categorization further implies that aleatoric and epistemic uncertainty—in addition to the extent to which they can be compartmentalized based upon their source—are uniquely defined based upon how decision makers attempt to handle them. This is to say, because internal and external-singular uncertainties are intrinsically framed in terms of an individual's conception of their truth values, decision makers can reduce both of them with acquisition of additional information or novel awareness of some pattern or causality, which inform their interpretation of whether said the associated events will or will not occur (C. R. Fox and Ülkümen, 2011). Conversely, the same does not hold for uncertainties in the external-distributional mode. For distributional events, any additional information that the decision maker obtains only contextualizes the relative propensity of events; it does not point to the truth value of a single event's occurrence in the same fundamental way that knowledge that reduces epistemic uncertainty does. As such, because decision makers cannot reduce aleatoric uncertainty, they must leverage their awareness of its relative existence to manage it (C. R. Fox and Ülkümen, 2011).

Representation of knowledge in probability theory

From a modeling standpoint, the attribution of internal and external-singular uncertainties to decision makers' epistemic states further ties into a shortcoming inherent in probability theory itself: the inability of probability distributions to represent a lack of knowledge, which can philosophically be conflated with the distribution's representation of the relative propensity of events. One commonly accepted example in statistics that demonstrates this failure is the use of the uniform distribution to represent complete ignorance in probabilistic terms. Although frequently adopted in modeling assumptions, this representation of ignorance is not entirely discriminative—a uniform distribution might also be used to represent a decision maker's complete knowledge that an event has perfectly equal probabilities for different outcomes and thus does not clearly reflect the decision maker's state of ignorance.

In general, this flaw of probability theory is intrinsically rooted in the inability to lessen the amount of knowledge that is contained in a distribution, particularly when compared to methods of learning that are *set-based* in nature (Hüllermeier and Waegeman, 2019). Version space learning, which is a logical approach to machine learning that searches a predefined hypothesis space, employs such an approach, expressing uncertainty in sets of candidate hypotheses and sets of candidate outcomes. Knowledge about the ground-truth is expressed in terms of a subset C within the total space of possible candidates; a larger set C corresponds to an increasing lack of knowledge and a smaller set C corresponds to increasing knowledge; as such, a common uncertainty measure that can be applied in such a context is the information-theoretic metric $\log(|C|)$ (Hüllermeier and Waegeman, 2019).

Furthermore, what inherently distinguishes set-based learning from probability theory is the simple relationship between the size of the set C and the epistemic state of the decision maker. Whereas in probability theory it is not possible to remove a candidate among the possible elements without changing the probability or propensity associated with all the other candidates, in set-based learning it *is* indeed possible to add or remove candidates without decreasing the plausibility of other candidates. That is to say, whereas the total amount of knowledge in a probability distribution remains fixed and is distributed among the possible candidates, it is variable in a set, in which candidates may remain equally plausible even as others are added or removed (Hüllermeier and Waegeman, 2019). By extension, because hypotheses are expressed exclusively in terms of being possible or not, all uncertainty in version space learning is considered epistemic; no aleatoric equivalent exists at all.

It is important to note that while there are other generalizations of probability theory that are better suited for uniquely representing epistemic uncertainty, among them imprecise probability (Walley, 1991), evidence theory (Shafer, 1976), and possibility theory (Dubois and Prade, 1988), systematically evaluating and benchmarking them is not the focus of this work. Rather, we introduce the notion of set-based representations as an alternative to probabilistic modeling from an instructive standpoint—at once to contextualize the difficulties of fully capturing uncertainty with probabilistic methods and to provide the philosophical language and background to understand the more technical approaches and metrics outlined later in this work.

1.2 Importance of uncertainty in decision-making

While the case for accurate uncertainty estimates in applications of deep learning is well-established in machine learning literature, there is considerably less emphasis on the need for accurate aleatoric and epistemic uncertainty estimates. This being said, however, if we examine the commonly cited motivating reasons for accurate uncertainty estimates, we find that they provide equal—if not greater—rationale for quantifying aleatoric and epistemic uncertainty as separate entities.

In general, it is maintained in the literature that uncertainty estimates are desirable in deep learning models that are used for decision making for a number of reasons:

1. They indicate when one should abstain from prediction or, at the very least, be hesitant or tentative in prediction. In fact, Tagasovska and Lopez-Paz (2019) notes that abstention is one common strategy to handle anomalies, outliers, out-of-distribution examples, and adversarial examples.

2. They imply that one needs to change the model due to the misspecification of model structure or assumptions, which is a commonly overlooked factor that contributes to overall uncertainty (Hüllermeier and Waegeman, 2019). In other words, when a model is not well specified, large uncertainty estimates might be reflective of high bias.

3. They provide insight into the structure of the noise, such as in the estimation of predictive intervals (Tagasovska and Lopez-Paz, 2019). In some cases, this might be so informative so as to indicate that it is worthwhile or necessary to seek out additional (or better) data to improve predictive accuracy.

4. They provide a step towards model interpretability and improve our understanding of the domains in which models are generally accurate, which is useful when considering how one might wish to deploy a model for real-life tasks (Tagasovska and Lopez-Paz, 2019).

However, if we consider these scenarios within the broader framework of aleatoric and epistemic uncertainty that we have established, we realize that accurate estimates of these two subtypes of uncertainty would be especially informative—and arguably more useful than a general estimate of total uncertainty. We discuss the advantage of having distinct estimates of aleatoric and epistemic uncertainty for each of these aforementioned cases, maintaining the same order as above:

1. Accurate aleatoric and epistemic uncertainty estimates equip models to identify inherently different causes for abstention. In the case of a predictive instance with high aleatoric uncertainty, for example, a model might suggest that it is unlikely—if not impossible—to improve predictive performance due to the inherent stochasticity of the case at hand; the data instance might correspond to a part of the domain where there is heavy class overlap or irreducible noise, for example. By comparison, in the case of high epistemic uncertainty, a model might suggest that prediction can be improved with more data or that an outside expert might supply the necessary additional knowledge to reduce uncertainty. Furthermore, distinguishing these reasons for abstention is useful for detection of anomalies, outliers, out-of-distribution examples, and adversarial examples, which might be based particularly in aleatoric or epistemic sources of uncertainty.

2. Uncertainty related to specification of model structure and model assumptions is normally thought to be epistemic in nature. Parameter uncertainty, for one, is considered to be directly related to the amount and quality of the available information (Der Kiureghian and Ditlevsen, 2009).

3. Depending on the dataset, noise might be rooted in aleatoric or epistemic uncertainty (or, in some cases, both). Noise attributed to high epistemic—but not aleatoric—uncertainty would suggest that one should seek out additional data.

4. Model interpretability by definition improves when decision makers can identify the distinct philosophical causes underlying the existence of uncertainty. Assessment of the domains in which models should be deployed, moreover, can be based in part upon identification of domains with low epistemic or aleatoric uncertainty, depending on the nature of the downstream task.

Thus, in all of the scenarios discussed above (and in any others that are not explicitly discussed here), having additional information about the decomposition of uncertainty into its aleatoric and epistemic components is strictly better for informed decision-making. To illustrate this point more concretely, we discuss a real-life setting in which control of decision-making is increasingly being ceded to automated systems: medical diagnosis. In this context, accurate aleatoric and epistemic uncertainty estimates have the potential to improve AI safety and prevent adverse outcomes.

Medical diagnosis

Models that are used for medical diagnosis, which are typically based upon assessment of electronic health records or medical images of some sort, attempt to automate detection of specific conditions or identification of high-risk patients. In the simplest sense, diagnostic models that output high uncertainty estimates for specific patients might opt to abstain from diagnosis and instead notify a physician, who can provide expert input. There is considerable rationale for this approach based upon Laves et al. (2019), who find that there is a correlation of $\rho = 0.99$ between prediction uncertainty and prediction error for computer-aided diagnosis (CAD) using deep learning for retinal optical coherence tomography (OCT) scans.

Moreover, decomposing uncertainty into its separate aleatoric and epistemic sources is valuable because it explains the extent to which uncertainty is rooted in the intrinsic difficulty of the diagnostic task or the size of the training data, as demonstrated by Tanno et al. (2019) for neuroimaging. This is particularly important for diagnostic applications because obtaining data in healthcare is difficult and intensive—patient privacy is of paramount importance, imaging technology is expensive, and rare diseases have few patients and even fewer datasets. As such, being able to pinpoint whether diagnostic uncertainty can be attributed to scarcity of data—i.e., whether it is epistemic—informs whether or not it is desirable to collect more data or data of a particular type that is especially informative. For example, understanding the areas in which epistemic uncertainty is highest might suggest where to focus data-collection efforts, e.g., for specific patient demographics or levels of severity of a disease.

1.3 Uncertainty in deep learning

The objective of this thesis is to identify, *evaluate*, and compare model-agnostic and flexible metrics for *quantifying* aleatoric and epistemic uncertainty in deep learning—as per the name of this work. In order to contextualize how we identify such metrics for quantifying uncertainty, it is important to first understand how uncertainty is generally treated in deep learning, which is the focus of this section.

First, it is critical to recognize the distinction between *modeling* uncertainty, *quantifying* uncertainty, and *evaluating* estimates for uncertainty. *Modeling* uncertainty refers to the way in which one formally represents uncertainty in a model or, in other words, how uncertainty is incorporated as an inherent part of the model structure. *Quantifying* uncertainty, by comparison, refers to the way in which we measure the amount of uncertainty there is in a model, which is typically done by applying and computing different metrics, which—as we will see—vary in their computational complexity. Approaches for quantifying uncertainty can be either model-specific or model-agnostic, and the models to which they are applied might *model* uncertainty in different ways. We note, however, that this thesis prioritizes approaches for quantifying uncertainty that are model-agnostic, which is considered desirable because such approaches can be more broadly and flexibly applied. Finally, *evaluating* uncertainty means assessing whether or not the amount of uncertainty estimated from our different approaches for modeling and quantifying uncertainty (which are usually paired together) is accurate, appropriate, or useful in light of the dataset and task at hand. Evaluation of uncertainty is typically done on other machine learning tasks such as reinforcement learning, active learning, and Bayesian optimization. More recently, work such as Gal (2016) and AngelosFilos, Gomez, and Rudner (n.d.) has attempted to evaluate uncertainty estimates directly for real domain-specific tasks.

Now motivated with this understanding, in this section we will first review the existing approaches that are used to model aleatoric and epistemic uncertainty in machine learning, noting that many of these approaches require significant adaptations to the conventional models in which we are most interested (which we will point out explicitly).

Then, having established these models, we will note that the literature lacks a systematic evaluation of general metrics that can be used to *quantify* aleatoric and epistemic uncertainty across many of these models. We will then proceed to highlight in §1.4 how this thesis will contribute to the literature: documenting and *evaluating* such uncertainty metrics that can be flexibly applied to a diversity of models and downstream tasks.

1.3.1 Modeling and quantifying aleatoric and epistemic uncertainty

At a high level, machine learning literature makes a few general assumptions about which sources of uncertainty are considered aleatoric or epistemic. For instance, aleatoric uncertainty is often thought of as noise that is learned by optimizing the per point model precision (Gal, 2016). On the other hand, uncertainty about the model itself (that is, uncertainty stemming from implicit assumptions about the model hypothesis space) and uncertainty about the weight parameters of the model are commonly understood to be epistemic and are computed using model averaging (Hüllermeier and Waegeman, 2019). Practically speaking, capturing epistemic uncertainty is typically predicated upon identifying regions of the input space that have little or no training data.

These assumptions aside, however, novel approaches for modeling uncertainty—which have varying levels of underlying justification and foundation in theory—appear in the literature quite regularly (Hüllermeier and Waegeman, 2019). That being said, however, many of said approaches fail to identify clearly which components of total uncertainty are aleatoric or epistemic—a tendency that reflects the overarching difficulty of clearly discriminating between the two sources in the model output.

Here, we review the broad range of approaches that are most commonly used for modeling *either* of the aleatoric and epistemic components of uncertainty. we note that all of these approaches are uniquely focused on just one of the two types of uncertainty; many therefore cannot be deployed with approaches that either model or provide an avenue for quantifying the complementary type of uncertainty in a unified, easy-to-implement, and model-agnostic framework. We accredit this review of approaches to Tagasovska and Lopez-Paz (2019).

Aleatoric uncertainty

As mentioned above, existing approaches for modeling aleatoric uncertainty are generally based upon the premise of learning about the per point conditional distribution of a target variable. There are a number of ways to approach this, as outlined by Tagasovska and Lopez-Paz (2019):

1. The conventional approach (which we review closely in §3.2.1) is to assume that the conditional distribution of the target variable is Gaussian. Per this assumption, one output layer of a neural network can be trained to model the variance of the Gaussian at each point (Kendall and Gal, 2017). Although this approach allows for neural networks to model noise as data-dependent and thus capture heteroscedastic (non-uniform) variance, the assumption that aleatoric noise is Gaussian means that non-symmetric, multimodal noise profiles cannot be accurately approximated.
2. A second approach involves the use of non-linear quantile regression, which seeks to learn the conditional quantiles of the target variable. Models that use quantile regression are either based upon decision trees (which do not fall into our category of interest) or attempt to train neural networks using pinball loss. Tagasovska and Lopez-Paz (2019) proposes an example of the latter category that explicitly links quantiles to aleatoric uncertainty and we review this model in §3.2.4 and implement it alongside the conventional approach described in the bullet above.
3. A third approach is to train neural networks using learning objectives with metrics that capture the quality of the prediction interval (PI), such as Mean Prediction Interval Width (MPIW) and Prediction Interval Coverage Probability (PICP) (Pearce, Zaki, Brintrup, and Neely, 2018). This produces accurate and high-quality PIs that capture per point variance, which is treated as aleatoric uncertainty.
4. A fourth approach entails implicit generative models, which define a stochastic procedure that directly generates data (Mohamed and Lakshminarayanan, 2016). Such models can output multiple predictions corresponding to an input. One can then use

this distribution of predictions to capture the aleatoric uncertainty, which—unlike the first approach—can reflect non-symmetric, multimodal noise.

Considering these strategies, however, we realize that each—with the exception of the first—requires substantial modifications to any existing, conventional model. The second and third models require adapting the loss function to incorporate a new learning objective—which may be incompatible with more complex models applied to real datasets—and the fourth is a distinct model class that is generally difficult to train and entails separate implementation altogether. For this reason, this thesis generally adopts use of the first approach, which adapts the loss function to model aleatoric variance as learned loss attenuation and only requires the addition of a corresponding output layer. While we consider a variant of the second approach known as Simultaneous Quantile Regression (SQR), which is used with basic, single-model NNs and has an available codebase (Tagasovska and Lopez-Paz, 2019), we largely consider this approach to be adjacent to our primary purpose. We leave evaluation of the remaining approaches for future work.

Epistemic uncertainty

Epistemic uncertainty is almost always modeled as variance that arises from the posterior distribution (§3.2.2). Given that the notion of sampling from the posterior distribution is so widely accepted in the literature, there are no other approaches for *modeling* epistemic uncertainty that we consider in this work. However, once we accept this model, there are a number of ways to *quantify* epistemic uncertainty:

1. The most traditional approach for quantifying epistemic uncertainty is generating a posterior predictive distribution, which we formally define in §2.2.1. At a high level, this involves sampling from the posterior distribution construct a distribution of possible unobserved values conditional on the values that are observed in the samples.

2. A second strategy for quantifying epistemic uncertainty is detection of out-of-distribution (OOD) examples (which is more typically applied to classification and has considerably less literature for regression). We review and implement several OOD detection approaches in §3.3.3. One more artisanal approach that some OOD methods adopt is to map in-domain examples to a constant value (the most common of which is zero) and out-of-distribution examples to other values, thereby signaling epistemic uncertainty. The Orthonormal Certificates (OCs) method proposed by Tagasovska and Lopez-Paz (2019), which we discuss in §3.3.3 could be considered an example of such an approach. Furthermore, Tagasovska and Lopez-Paz (2019) additionally notes that anomaly and

outlier detection and one-class classification might also be framed as attempts to quantify epistemic uncertainty, although these approaches are not an area of focus in this thesis and are left for evaluation in future work.

3. One last category of approaches, which encompasses noise-contrastive priors (Hafner et al., 2018) and generative adversarial networks (GANs) (Goodfellow, Pouget-Abadie, et al., 2014), entails using data to construct realistic "negative examples" outside of the training data distribution. A predictor trained to identify such negative examples could then estimate the epistemic uncertainty associated with specific inputs.

Each of the above strategies varies in its adaptability and ease of application to existing models. The first approach is considered universal, as it arises as an extension of Bayesian modeling. It can, however, be easily extended to frequentist models as well. While it is the basis of comparison for the treatment of epistemic uncertainty in this thesis, it is important to note that repeatedly sampling from a model as specified above is likely to capture not only epistemic but also aleatoric variance. Given this, it becomes desirable to derive a decomposition that can uniquely distinguish aleatoric and epistemic variance within the posterior predictive. This is a recurring theme within this work.

Furthermore, while the second category of strategies—OOD detection—represents a wide range of literature, many OOD detection approaches can be applied to the last layer or output layer of a previously trained neural network; they are therefore relatively simple to implement and, for this reason, are included in this thesis.

Finally, the last category of strategies, which take on a generative approach, is the least straightforward to implement on top of existing architectures. GANs are a distinct model class requiring separate implementation and noise-contrastive priors entail the use of an input prior, which adds noise to the inputs, and an output prior, which is a wide distribution given these inputs (Hafner et al., 2018).

1.4 Thesis contributions and structure

Given the different approaches that exist for modeling aleatoric and epistemic uncertainty and the lack of existing metrics that formalize desiderata for quality estimates of aleatoric and epistemic uncertainty, it is clearly important to identify metrics for *quantifying* aleatoric and epistemic uncertainty that are model-agnostic and can be applied on top of the different modeling approaches discussed above.

Following this line of reasoning, the following areas for contribution—with respect to *quantifying* uncertainty—become evident:

1. First, the literature lacks a systematic *evaluation* and comparison of the existing metrics for quantifying aleatoric and epistemic uncertainty that indeed are model-agnostic and easy to implement. The primary goal of this thesis is therefore to gather and *evaluate* such metrics for both regression and classification tasks. The main metrics in which we are interested arise from three categories: statistical measures of variability in categorical distributions (as reviewed in §3.3.1), *decompositions* of aleatoric and epistemic uncertainty (which we discuss in the third bullet point below), and out-of-distribution detection metrics, which are proxies for epistemic uncertainty. Historically speaking, these different categories of metrics have not been evaluated comparatively, and this thesis is novel in its attempt to evaluate them under one framework.

2. Following the above point, the literature also lacks a systematic *evaluation* of how metrics for quantifying aleatoric and epistemic uncertainty fare when paired with different models and their accompanying inference methods (the number of which has rapidly increased in recent years). As such, one of the supporting objectives of this work is to evaluate the quality of uncertainty metrics when coupled with seven different methods (which we review in §4) and the models that they approximate, which are carefully selected to include approaches rooted in both frequentist and Bayesian perspectives.

3. Machine learning literature would benefit from more *decompositions* of aleatoric and epistemic uncertainty—i.e., ways to *quantify* aleatoric and epistemic uncertainty from estimates of total uncertainty that are model-agnostic, easy to implement, and mathematically or statistically principled in some way. Such decompositions would ideally treat these two uncertainties as contributing parts of total uncertainty. In §3.3.2, we review the existing decompositions that satisfy these criteria and propose extensions of these decompositions, although we acknowledge that derivation of novel, principled decompositions of uncertainty is rather difficult.[1]

We find a number of interesting results for both regression and classification. First, we find that metrics based upon neural networks' last layer representations of uncertainty provide an attractive and interesting alternative for quantifying epistemic uncertainty for regression tasks, particularly given that conventional approaches for regression do not consider such metrics or OOD detection approaches, which tend to be reserved for classification in general. This suggests an exciting avenue for future research.

We obtain accurate estimates for aleatoric and epistemic uncertainty by applying existing decompositions of uncertainty, as proposed by Depeweg et al. (2017), in novel ways. For

[1] In light of our preceding discussion on different representations of knowledge, we note that, for the purposes of this thesis, the decompositions we consider are based in the use of probability distributions, which we accept—due to their widespread use across literature—as convention for uncertainty quantification.

one, we find that the law of total variance, when paired with deep ensembles of probabilistic NNs, provides accurate estimates for both aleatoric and epistemic uncertainty for regression tasks that outperforms existing models, namely MC dropout. We also adapt an existing statistical decomposition to propose a way to decompose any statistical metric of variability for categorical distributions into its aleatoric and epistemic components, which we evaluate and find to be accurate—and consistent with estimates provided by predictive entropy and mutual information—for classification.

Finally, with regards to models, we find that the neural linear model, which places priors only on the last layer of a NN, provides a tractable alternative to Hamiltonian Monte Carlo (which is considered to be a means of obtaining the ground truth for the posterior of Bayesian NNs) with comparable uncertainty estimates. We additionally find that MC dropout does not produce high-quality uncertainty estimates. We evaluate the decomposition of aleatoric and epistemic uncertainty proposed in Kendall and Gal (2017) to find that it is inconsistent, inaccurate, and highly dependent on tuning the dropout rate, which perhaps suggests that we should opt for alternative inference methods for heteroscedastic models.

1.4.1 Thesis structure

The first half of this thesis (§2–4) is focused on exposition. §2 provides the necessary background on traditional deep learning and Bayesian deep learning. §3 describes different ways of modeling aleatoric and epistemic uncertainty and reviews the different metrics use to quantify aleatoric and epistemic uncertainty that we will evaluate in this work. §4 describes the different models and accompanying inference methods that we will consider.

The second half of this thesis (§5–7) is focused on evaluation. §5 evaluates the metrics and methods that we review in the expository half of the thesis on synthetic (i.e., toy) data for both regression and classification. §6 does the same but on real datasets. Finally, §7 presents conclusions from the experiments in the preceding two sections and proposes areas for future research.

Chapter Two

Background on Deep Learning

As mentioned in the introduction, the field of deep learning is a subset of machine learning that is built upon the use of collections of connected computational units known as artificial neural networks (NNs). In this field, the use of the word "deep" implies that the neural networks in question have many layers of computational units, which is loosely interpreted in the literature as indicating at least two layers.

In this chapter, we outline two different classes of approaches for constructing and training NNs, both of which are used in this thesis. The first class, described in §2.1, is the traditional view of NNs, which relies upon point estimates of model parameters and forms the backbone of traditional deep learning. The second class, described in §2.2, is the Bayesian view of NNs, which models NNs probabilistically within the framework provided by Bayes' rule and forms the backbone of Bayesian deep learning (BDL).

This section is not exhaustive in reviewing the field and primarily introduces the aspects of deep learning that are necessary to understand this thesis. Please note that we accredit Bishop (2006), Gal (2016), and Moberg (2019) for inspiring these explanation of both neural networks and Bayesian neural networks. For a more extensive introduction to the field, we direct the reader to Bishop (2006), Goodfellow, Bengio, and Courville (2016), and Murphy (2012), the last of which has a special emphasis on BDL.

2.1 Neural networks

At the highest level, neural networks, which can be used for both regression and classification tasks, can be thought of as a multi-layer extension of linear and logistic regression, respectively.

Let us define a set of N observations of inputs and outputs $\{(\mathbf{x}_1, \mathrm{y}_1),..., (\mathbf{x}_N, \mathrm{y}_N)\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^Q$. We wish to apply some transformation $\mathbf{f}(\mathbf{x}) = \hat{\mathbf{y}}$ from the N by D matrix $\mathbf{x}$ to the N by P matrix $\hat{\mathbf{y}}$, which is an approximation for $\mathbf{y}$.

A feed-forward neural network, which is visualized in Figure 2.1 and can be conceived

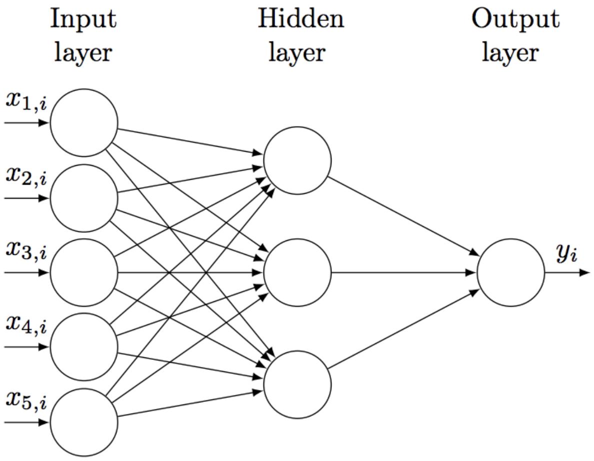


Figure 2.1 A two layer feedforward neural network with input dimension 5 and output dimension 1.

as such a function $\mathbf{f}(\mathbf{x})$ that is parameterized with $\mathbf{W} \in \mathbb{R}^D$ and some bias $\mathrm{b} \in \mathbb{R}^Q$, seeks to approximate this output $\mathbf{y}$ given some input $\mathbf{x}$. It can be thought as a series of linear transformations of the form $\mathbf{x}\mathbf{W}_i + \mathrm{b}_i$ from $\mathbf{x} \in \mathbb{R}^D$ to $\mathbf{y} \in \mathbb{R}^Q$ with the following properties:

1. Neural networks include nonlinear transformations on top of the linear transformations represented by $\mathbf{W}_i$ and b_i. Each layer of the neural network applies some nonlinear, differentiable activation function $\sigma(\cdot)$ to the output of the linear transformation $\mathbf{x}\mathbf{W}_i + \mathbf{b}_i$ in that layer, such that each layer can be represented with the nonlinear transformation $\sigma\left(\mathbf{x}\mathbf{W}_i + \mathbf{b}_i\right)$. Commonly used activation functions include the rectified linear unit (ReLU)[1] and sigmoid.[2]

2. Neural networks apply multiple layers of these nonlinear transformations iteratively. In other words, the nonlinear transformation of the j-th layer is applied to the output of the $(j-1)$-th layer for all layers of the neural network with the exception of the first layer, which is known as the input layer and is applied to $\mathbf{x}$. Moreover, the last layer is known as the output layer, and the layers in between the input and output layers are known as hidden layers.

Then, for example, a two-layer neural network applied to some input $\mathbf{x}$ can be defined with the function $\mathbf{f}(\mathbf{x})$, where

$$\mathbf{f}(\mathbf{x}) = \sigma_2\left(\left(\sigma_1\left(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1\right)\right)\mathbf{W}_2 + \mathbf{b}_2\right) = \hat{\mathbf{y}}.$$

[1] $\mathrm{ReLU}(x) = \max(x, 0)$.

[2] $\sigma(x) = (1 + \exp(-x))^{-1}$.

In order to find the optimal weights $\mathbf{W}$ and bias b for this function $\mathbf{f}(\mathbf{x})$, we want to minimize some loss function $\mathcal{L}(\mathbf{W}, \mathbf{b})$ with respect to $\mathbf{W}, \mathrm{b}$ between the predictions $\hat{\mathbf{y}}$ and true output $\mathbf{y}$ (which we discuss further in §2.1.1).

Remembering that $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{y} \in \mathbb{R}^Q$ in our introduction, it then follows that $\mathbf{W}_1$ must have dimensionality of D by P, $\mathbf{b}_1$ must be a vector of size P, $\mathbf{W}_1$ must have dimensionality of P by Q, and $\mathbf{b}_2$ must be a vector of size Q. Extending these principles to a neural network with k layers, we note that $\mathbf{W}_1$ must have D rows when $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{W}_k$ must have P columns when $\mathbf{y} \in \mathbb{R}^P$. Meanwhile, the dimensions of the hidden layers can vary but must be such that each layer can feed into the next. It is also typical for the last bias $\mathbf{b}_k$, which in this case would be $\mathbf{b}_2$, to be set to zero. Furthermore, in the case of regression, it is common for the last activation function $\sigma_k(\cdot)$ to provide a linear output without any nonlinear transformation. In this case, σ_2 would simply output $(\sigma_1 (\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1))\, \mathbf{W}_2 + \mathbf{b}_2$.

By comparison, in the case of classification, we set the last activation function σ_k to be a softmax function. The softmax function outputs the respective probabilities of an input being classified with a label in the set $\{1, ..., C\}$. It can be defined as

$$\sigma(\hat{y}_c) = \frac{\exp(\hat{y}_c)}{\sum_{c'} \exp(\hat{y}_{c'})} = \hat{p}_c, \tag{2.1}$$

where $\hat{y}$ is the C-dimensional last output of the NN prior to this sigmoid function, c denotes the label for the desired class, c' denotes the set of all the labels, and $\hat{p}_c$ is a C-dimensional vector containing the probabilities of an input belonging to each class. Furthermore, we note that in the simplest case of classification (a binary classification task in which our NN only has one layer with parameters $\mathbf{W}_1, \mathrm{b}_1$), the function $\mathbf{f}(\mathbf{x})$ becomes equivalent to binary logistic regression, as the softmax function is equivalent to the sigmoid function when there are only two classes.

2.1.1 Loss functions and training

Broadly speaking, the goal of optimizing the parameters $\mathbf{W}, \mathrm{b}$ of a neural network regression model with respect to the training data $\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}$ is to produce a model that can generalize to some unobserved test data $\mathbf{X}_{\text{test}}, \mathbf{Y}_{\text{test}}$.In order to find the best approximation of $\mathbf{y}$, we optimize the neural network by choosing parameters $\mathbf{W}$ and $\mathbf{b}$ across all k layers of the neural network to minimize a specified loss function.

The most common loss function used for regression is the *mean-squared error* (MSE) between the predictions $\hat{\mathbf{y}}$ and true output $\mathbf{y}$, defined as

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^{N} (\mathrm{y}_i - \hat{\mathrm{y}}_i)^2, \tag{2.2}$$

where $\hat{\mathrm{y}}_i$ is the prediction given by the function $\mathbf{f}$ for the i-th observation in the data.

Loss functions differ in the cases of regression and classification. Neural networks that are used for classification tasks typically seek to minimize a *cross-entropy* loss with respect to $\mathbf{W}, \mathbf{b}$. The cross-entropy loss takes the negative log of the probabilities predicted for the observed label, as output by equation (2.7), and is defined as

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = -\frac{1}{N} \sum_{i=1}^{N} \log \hat{p}_{i,c_i}, \tag{2.3}$$

where c_i is the observed class label for input i and $\hat{p}_{i,c_i}$ is the prediction of the neural network for input i corresponding to the specified label c_i.

Note that the cross-entropy loss is the *negative log likelihood* (NLL) of the multinomial distribution, where negative log likelihood is another commonly used cost function that we will later reference in §3. Note that NLL is a popular cost function and is often used to obtain maximum likelihood estimates (MLE) in statistical models because maximizing likelihood is equivalent to minimizing NLL.[3]

Modeling distributions

By modifying the loss function to be variance-dependent, we can construct a neural network that outputs estimates for the mean and variance of a normal distribution, such that $\mathbf{f}(\mathbf{x}) = (\hat{\mu}, \hat{\sigma})$ is the output of the NN and we can model $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$. We select the NLL of the normal distribution as the loss function,

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2} \log \hat{\sigma}_i^2 + \frac{(y_i - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2} \right), \tag{2.4}$$

where we construct the NN to output $\log \hat{\sigma}_i$ for numerical stability and transform it as necessary for the loss function.

Such a neural network, although not probabilistic in the same sense as a Bayesian NN (described in §2.2), provides one means of modeling uncertainty homoscedastically. While it is possible to use other distributions in these pseudo-probabilistic NNs, we use a normal distribution out of convention. We use such neural networks to construct deep ensembles, as described in §4.1.2.

[3] Most computational frameworks are designed to minimize rather than maximize functions, so we take the negative of the likelihood to suit this constraint. Moreover, minimizing log likelihood is equivalent to minimizing likelihood due to the monotonically increasing nature of the log function. Lastly, NLL is used because applying the log function allows for simpler analytical representation of the likelihood and for more stable numerical computation.

Learning process

NN weights are typically initialized randomly and optimized using gradient-based algorithms that minimize the specified loss function with respect to NN parameters; this learning process is referred to as *training* the NN and can be viewed as a global optimization problem. Traditional gradient descent algorithms compute the gradient of the loss function with respect to $\mathbf{W}$ and $\mathbf{b}$ (typically using backpropagation as a gradient computing technique) and take small steps, the size of which is defined by a specified learning rate η, in the opposite direction of the gradient. With a sufficient number of steps, the loss eventually converges to some local minimum.

However, because loss landscapes for neural networks are not strictly convex, there are typically many local minima, all of which but one—the global minimum—do not reflect the optimal model parameters $\mathbf{W}$ and $\mathbf{b}$. As such, because local optimization parameters such as gradient descent are not equipped to find the global minimum of the loss function, it is more common to use gradient descent algorithms that stochastically approximate the gradient and thus introduce noise into the gradient descent path. For instance, mini-batch gradient descent, also known as stochastic gradient descent (SGD),[4] uses the same update rule as traditional gradient descent but approximates the gradient on some small mini-batch of the data. Resultantly, using mini-batch gradient descent, especially with small batch sizes, means that the loss will eventually decrease over time, but that individual update steps will not necessarily shift the loss in the optimal direction, which allows the algorithm to escape local minima.

2.1.2 Regularization

Because NNs do not require many assumptions—e.g., linearity—about the systems that they are approximating, they can be applied almost universally and are appropriate for tasks that lack clearly documented, functional forms. In fact, previous work has demonstrated that NN architectures can approximate any continuous function as long as a sufficiently large hidden layer is constructed, and the simple feedforward architecture described earlier can be adapted to more specialized architectures designed to handle image or text inputs, making NNs as a model class deeply flexible.

However, the high expressiveness of NNs makes them deeply prone to overfitting to the training data $\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}$, which limits their ability to generalize well to unseen data. In order to prevent this phenomenon, it is typical to use *regularization* techniques, the most common of which are weight decay and dropout.

[4] Stochastic gradient descent technically refers to mini-batch gradient descent with a batch size of one, but the two terms are often used interchangeably.

Weight decay

Weight decay is applied to NNs by adding a term to the loss function that penalizes the norm of the weights with the goal of encouraging the weights to be smaller. The two most common types of weight decay are L_1 and L_2 regularization, which penalize the absolute value norm and squared norm of $\mathbf{W}$ and $\mathbf{b}$ respectively. Applying L_2 regularization with some regularization rate λ for the weights $\mathbf{W}$ and bias $\mathbf{b}$ in a k-layer NN would update the learning objective for regression to

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2 + \lambda\sum_{j=1}^{k}||\mathbf{W}_j||^2 + \lambda\sum_{j=1}^{k}||\mathbf{b}_j||^2,$$

where it is also possible to introduce separate rates λ_j for the parameters $\mathbf{W}_j$ and $\mathbf{b}_j$ on different layers of the NN.

Dropout

When applying dropout as a regularization technique, the output of individual NN units is retained with some specified probability $p \in (0, 1)$ and otherwise set to zero (Srivastava et al., 2014). Units (and their incoming and outgoing connections) are be randomly dropped in one, several, or all layers with the intent of preventing units from co-adapting excessively. Dropout is typically applied during the training process, which can be thought of as comparable to training an ensemble of thinned networks that adapt to the data in different ways. Then, during inference, dropout is turned off and predictions are obtained from an unthinned network with smaller weights, which improves predictive performance when generalizing to unseen data.

One approach known as Monte Carlo dropout (Gal and Ghahramani, 2016) that we will review in §2.2.3 and implement as described in §4.2.3 turns dropout on during the inference process to model uncertainty in what is claimed to be a Bayesian approximation.

2.2 Bayesian neural networks

Whereas traditional neural networks are trained via gradient descent to provide a deterministic output given some input $\mathbf{x}$ (or a deterministic output for the mean and variance of a normal distribution), Bayesian neural networks (BNNs) are neural networks that place a prior distribution over their weights $\mathbf{W}$ (Neal, 2012) and provide a way to represent the parameter space probabilistically. As such, BNNs are considered to be state-of-the-art for modeling uncertainty.

Although exact inference for BNNs is intractable due to the complicated nature of the posterior landscape, advances in approximate inference approaches have made BNNs more computationally practical. In this section, we first review the foundations of Bayesian inference in §2.2.1, describe how neural networks are constructed within the Bayesian framework in §2.2.2, and finally review approximate inference techniques for BNNs in §2.2.3.

2.2.1 Bayesian inference

Let us once again consider a set of N observations of training data $\{\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}\} = \{(\mathbf{x}_1, \mathrm{y}_1),..., (\mathbf{x}_N, \mathrm{y}_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^P$, that is generated by some underlying statistical process. Within the Bayesian inference framework, suppose that we can approximate this underlying data-generating process with some function $\mathbf{y} = \mathbf{f}^\theta(\mathbf{x}) + \epsilon$, where the true value of the parameter $\theta \in \Theta$ is unknown and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with some variance σ^2. Further supposing that we have some pre-existing knowledge about θ, we can represent our belief about the value of θ before any data is observed with some prior distribution $p(\theta)$. Then, ws define the likelihood function $p(\mathbf{Y}|\mathbf{X}, \theta)$ that represents how the outputs are generated from the inputs based upon our proposed probabilistic model $\mathbf{y} = \mathbf{f}^\theta(\mathbf{x}) + \epsilon$, which is parameterized by θ. In this vein, we can think of the likelihood as representing the likeliness that a model with some specific parameters θ generated our data (where Gaussian likelihood is typically used for regression and softmax likelihood for classification).

Using the prior and the likelihood, we compute a posterior distribution describing the probability of $\theta \in \Theta$ according to Bayes' rule:

$$p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{Y}|\mathbf{X})}$$

The posterior distribution represents our belief about the value of θ after data is observed, and analytically or approximately computing it is at the heart of Bayesian inference. Indeed, in order to make a prediction about the distribution of an output y^* corresponding to some new input x^*, we marginalize over the posterior to obtain the predictive posterior,

$$p(y^*|x^*, \mathbf{X}, \mathbf{Y}) = \int p(y^*|x^*, \theta)p(\theta|\mathbf{X}, \mathbf{Y})d\theta.$$

Because this marginalization often cannot be done analytically, it is common to sample θ from $p(\theta|\mathbf{X}, \mathbf{Y})$ and approximate the posterior predictive:

$$\begin{aligned} p(y^*|x^*, \mathbf{X}, \mathbf{Y}) &= \mathbb{E}_{p(\theta|\mathbf{X}, \mathbf{Y})}\left[p(y^*|x^*, \theta)\right] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} p(y^*|x^*, \theta_s), \quad \theta_s \sim p(\theta|\mathbf{X}, \mathbf{Y}) \end{aligned}$$

Another component of the posterior that is core to the inference process is the denominator, also known as the *marginal likelihood* or *model evidence*:

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)d\theta.$$

Although analytically computing the marginal likelihood is feasible for simple models such as Bayesian regression, the marginal likelihood is high-dimensional and difficult to compute—or intractable altogether—for models that are even somewhat more complicated. As a result, the posterior is often expressed as proportional to the prior and likelihood,

$$p(\theta|\mathbf{X}, \mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta),$$

It then becomes necessary to compute the posterior through approximate inference approaches, and the marginal likelihood $p(\mathbf{Y}|\mathbf{X})$, which is not dependent on θ, is treated as a normalizing constant. This is indeed the case for neural networks and the primary constraint limiting the practicality of BNNs. We discuss the most common methods for approximate inference in section §2.2.3 after describing the structure of BNNs in §2.2.2 and further contextualizing the difficulty of accurate inference.

2.2.2 Bayesian neural networks

Bayesian neural networks (BNNs) are constructed by placing prior distributions on the weights in the general neural network framework in §2.1. Likelihood functions are selected per the typical Bayesian approach, with Gaussian likelihood for regression and softmax likelihood for classification.

Given this, a BNN model for regression can be defined with

$$p(\mathbf{W}) = \mathcal{N}(0, \sigma^2_{\mathbf{W}}\mathbf{I}) \tag{2.5}$$

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}) = \mathcal{N}(\mathbf{y}; \mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma^2_{\mathbf{y}}\mathbf{I}), \tag{2.6}$$

where σ^2_W is a scalar selected as the variance for the Gaussian prior, $\mathbf{f}^{\mathbf{W}}(\mathbf{x})$ is the output of the neural network for an input $\mathbf{x}$ given $\mathbf{W}$, and $\sigma^2_{\mathbf{y}}$ is the model precision, which is also commonly regarded as an estimate for aleatoric variance (to be discussed in §3.2.1).

By comparison, a BNN model for classification would retain the same prior $p(\mathbf{w})$, but the likelihood would instead be

$$p(\mathbf{y} = c|\mathbf{x}, \mathbf{W}) = \frac{\exp\left(\mathbf{f}^{\mathbf{W}}_c(\mathbf{x})\right)}{\sum_{c'} \exp\left(\mathbf{f}^{\mathbf{W}}_{c'}(\mathbf{x})\right)}, \tag{2.7}$$

where c denotes the label for the desired class and $c' \in \{1, ..., C\}$ denotes the set of all possible labels.

Although expressing the model for a BNN is relatively straightforward, deriving an exact posterior—which is proportional to hundreds, if not thousands, of priors on the network weights and a likelihood function based upon those weights—is impossible. In fact, Skorokhodov and Burtsev (2019) demonstrated that BNN posteriors are so complex that it is possible to find two-dimensional projections that replicate any desired pattern therein. As such, the primary practical challenge of using BNNs is using approximate inference methods that are at once reliable in obtaining accurate representations of the posterior and scalable to networks used for real-life applications.

2.2.3 Approximate inference techniques

"Approximate inference" in deep learning refers to the optimization process that is used to approximate the otherwise intractable integration over model parameters. Broadly speaking, traditional approximate inference for BNN posteriors can be categorized into two overarching types of approaches: Markov chain Monte Carlo (MCMC) methods and variational inference (VI). Although MCMC methods are considered a "gold standard" for inference, they do not scale well to large datasets; as such, modern approximate inference relies primarily upon recent developments in VI.

One additional category of methods encompasses Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) and Bayesian ensemble methods (Pearce, Zaki, Brintrup, Anastassacos, et al., 2018), which have been proposed as approximate Bayesian inference approaches. Because these approaches do not attempt to approximate the posterior of BNNs directly as MCMC and VI do and instead apply stochastic sampling to traditional NNs, we consider them "semi-Bayesian" for the purposes of this thesis.

We broadly survey all three types of approximate inference techniques in this section. We then dedicate §4.2 to describing the mathematical details of the methods specifically used in this thesis, which include one from each of the three categories of approximate inference (HMC, BBB, and MC dropout).

Markov chain Monte Carlo methods

MCMC methods comprise a class of algorithms that sample from a distribution $p(\theta)$ by constructing a Markov chain that has p as its unique equilibrium distribution (equilibrium meaning that distribution satisfies *stationary* and *limiting* properties) and sampling from the chain. While MCMC methods are commonly used for approximation of intractable integrals, they are often highly inefficient in high dimensions, in which such samplers struggle to locate areas of high mass in the target distribution p.

Hamiltonian Monte Carlo (HMC) is one MCMC method that offers a solution to this problem by incorporating principles from Hamiltonian dynamics (Neal et al., 2011). By introducing a Hamiltonian dynamical system of potential energy, kinetic energy, and an added set of momentum variables, HMC avoids the slow exploration of the state space by proposing moves to distant states that are not correlated with the current state. This increases the probability of acceptance for those proposals and requires fewer Markov Chain samples for accurate approximation. However, while HMC is often cited as the ground truth for inference in BNNs and is able to rapidly explore state spaces, it is majorly limited by slow computation of the gradient of the potential energy function, which is needed to simulate Hamiltonian dynamics, and suffers from its inability to scale to larger networks.

As a potential solution to HMC's lack of scalability, Chen, E. Fox, and Guestrin (2014) marry the notion of stochastic mini-batch gradient computation with Hamiltonian dynamics to propose stochastic gradient HMC (SGHMC). SGHMC is an efficient stochastic approach to Bayesian posterior sampling that adds a friction term to the momentum update in order to retain stationarity at the target distribution. However, while SGHMC has payoffs in its improved speed and explorative sampling, which are advantageous when approximating complex, real-world distributions, the algorithm is not without tradeoffs. Because the added friction parameter does not have existing heuristics and is not easily defined, it complicates tuning and slows sampling, resulting in high burn-in times and demanding more precise parameterizations for the algorithm. Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011), another mini-batched version of HMC, and cyclical Stochastic Gradient MCMC (R. Zhang et al., 2019), a generalization of SGHMC and SGLD that uses a cyclical learning rate schedule, suffer from similar issues and are not theoretically guaranteed to converge when the gradient noise is not well-estimated. For this reason, we limit the use of MCMC methods in this thesis to HMC (which we review more closely in §4.2.1).

Variational inference

Because sampling methods are computationally expensive and slow to converge, there has been extensive emphasis in the literature on developing variational methods as a more practical alternative.

At a high level, variational methods attempt to approximate a target posterior distribution. Given some target posterior distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ where $\mathbf{W} \in \mathbb{R}^I$, we want to find some variational distribution $q_\theta(\mathbf{W})$, parameterized by θ, that best approximates p. This in turn requires selecting a family of variational distributions for approximating p and a divergence measure to quantify the difference between p and q, which is typically selected to be Kullback-Leiber (KL) divergence (Kullback and Leibler, 1951). The procedure known as *variational inference* entails the process of minimizing this divergence to optimize q.

The first attempts at variationally approximating the posterior of BNNs (Hinton and Van Camp, 1993; Barber and Bishop, 1998) relied upon a fully factorized approximation of $q_\theta(\mathbf{W})$. While such methods form the foundation for more modern variational methods, they scaled poorly due to difficulties in optimization that arose as a consequence of the intractable log likelihood in multi-layer BNNs.

In response to this limitation, Graves (2011) proposed the idea of applying stochastic sub-sampling techniques to the fully factorized approximation of q, which provided a way to apply VI at scale. Blundell et al. (2015) further improved upon the gradient computation described in Graves (2011) by proposing a method known as Bayes by Backprop (BBB), which approximates the posterior with diagonal Gaussian distributions and uses a trick to reparametrize the log likelihoood that was originally proposed in Kingma and Welling (2013). Although the fully factorized Gaussian variational family used in BBB fails to capture correlation among the weights in the posterior, we nevertheless implement and use BBB in this thesis, as it is a standard for variational methods in the field. We further discuss the technical details of BBB in §4.2.2).

There is a number of works that attempt to improve upon BBB. Probabilistic Backpropagation (PBP) (Hernández-Lobato and Adams, 2015), Black-box α-Divergence (BB-α) (Hernández-Lobato and Adams, 2015), and functional variational BNNs (fvBNN) (Sun et al., 2019) seek to capture important properties of the posterior distribution by using richer families of divergence measures (Yao et al., 2019). Matrix Gaussian Variate priors (MVG) (Louizos and Welling, 2016), Multiplicative Normalizing Flows (MNF) (Louizos and Welling, 2017), Bayes by Hypernet (BbH) (Pawlowski et al., 2017), and Noisy Kronecker-factored Approximate Curvature (K-FAC) (G. Zhang et al., 2017) use structured variational families with the intent of better capturing correlation in the posterior. As this work is not specially focused on evaluating different variational methods, we do not implement these methods.

Monte Carlo dropout and Bayesian ensembling

Although dropout was originally proposed as a regularization method for neural networks (§2.1.2), Gal and Ghahramani (2016) suggested turning on dropout during test time for traditional NNs. Per this protocol, many stochastic forward passes sample from what is effectively a distribution over the weights of the NN in an ensemble-like approach, thereby creating a distribution for the posterior predictive from which it becomes possible to obtain uncertainty estimates. We refer to this inference process as *Monte Carlo dropout*.

Although the mathematical premise underlying MC dropout is that the dropout objective minimizes the KL divergence between the approximate distribution and the posterior of a deep Gaussian process (Gal and Ghahramani, 2016), critics have pointed out flaws with MC dropout. One weakness of MC dropout is the need to tune the dropout rate p, which typically requires a grid search. Moreover, because the dropout rate does not depend on the data, the posterior predictive distribution is invariant to duplicates of the dataset (Osband, Aslanides, and Cassirer, 2018). As such, the posterior achieved via MC dropout does not concentrate asymptotically and, thus, some have claimed that MC dropout is not a true Bayesian approximation (Osband, 2016). For this reason, we classify it as "semi-Bayesian," as mentioned earlier in this section. That being said, however, we acknowledge that MC dropout is easy to implement and to scale, particularly when compared to other, more complicated approximate inference methods. As such, because it is one of the most commonly used approaches for obtaining uncertainty estimates, we use it as a staple in this thesis and review it more closely in §4.2.3.

Finally, although ensemble methods are generally considered to be frequentist, Pearce, Zaki, Brintrup, Anastassacos, et al. (2018) propose a way to modify traditional ensemble approaches by regularizing parameters about values drawn from the prior distribution, which they consider to be an *anchor distribution*. This approach, which is correspondingly called *anchored ensembling*, corresponds to a Bayesian inference category known as randomized MAP sampling (RMS). However, the authors note that anchored ensembling requires two special conditions—perfectly correlated parameters and extrapolation parameters—for theoretical guarantee that the method will recover the posterior, implying that the basis for applying RMS to NNs may be questionable. Although this thesis does not use Bayesian ensembling, we implement two types of ensembles (which are described in §4.1) as two non-Bayesian alternatives for obtaining uncertainty estimates.

Chapter Three

Uncertainty in Deep Learning

Now equipped with an understanding of traditional and Bayesian deep learning, we will review the different approaches for *modeling* and *quantifying* aleatoric and epistemic uncertainty that we *evaluate* in this thesis. We discussed in the introduction the differences between *modeling* uncertainty (i.e., the way we represent uncertainty as an inherent part of the model) and *quantifying* uncertainty (i.e., the way in which we measure the amount of uncertainty there is in a model). We note that this chapter is not exhaustive but, rather, outlines only the approaches for modeling and quantifying uncertainty that are included in this thesis. We refer the reader to §1.3 for a more high-level overview of how aleatoric and epistemic uncertainty are treated in deep learning.

In order to contextualize our discussion about modeling and quantifying aleatoric and epistemic uncertainty, §3.1 first discusses the language surrounding related types of uncertainty that arise in deep learning. Then, §3.2 reviews the most common approaches used for modeling aleatoric and epistemic uncertainty; we note that all of these approaches were highlighted in §1.3. §3.3 reviews different approaches for quantifying aleatoric and epistemic uncertainty, which includes classical statistical measures of variability that can be applied to discrete distributions to estimate total uncertainty, "decompositions," (principled ways of estimating aleatoric and epistemic uncertainty as complementary entities), and out-of-distribution detection metrics. We note that none of the classical statistical measures of variability are expressly intended to capture aleatoric or epistemic uncertainty, but we evaluate their use for classification, as they are flexible and easy to implement on top of existing models and, to our knowledge, have not been systematically applied in the context of deep learning.

3.1 Types of uncertainty

Taxonomies of uncertainty in deep learning literature often consider other types of uncertainty in addition to aleatoric and epistemic uncertainty, many of which are semantically related. Among the most commonly referenced types of uncertainty are *data uncertainty, model uncertainty, parameter uncertainty, in-between uncertainty, distributional uncertainty*, and *approximation uncertainty*. We can map these terms to our existing framework of aleatoric and epistemic uncertainty.

Data uncertainty pertains to uncertainty that arises from inherent properties of the data and describes how uncertain the relationship between two variables $\mathbf{X}$ and $\mathbf{Y}$ is, which might be attributed to class overlap, label noise, and homoscedastic and heteroscedastic noise (to be defined shortly in §3.2.1). Data uncertainty cannot be reduced with additional data and can be considered equivalent to aleatoric uncertainty (Malinin and Gales, 2018).

Model uncertainty and *parameter uncertainty* are closely related and often conflated but have important underlying differences. Model uncertainty pertains to the uncertainty about how well a model is matched to the data, whereas parameter uncertainty pertains to the uncertainty about the parameters of a model given some training data. Both can be reduced with additional training data and can be considered types of epistemic uncertainty (Der Kiureghian and Ditlevsen, 2009). Although it is important to recognize that the distribution over weights in a neural network is epistemic, this thesis is primarily interested in functionally computing estimates for epistemic uncertainty for prediction and, as such, does not focus in depth on the uncertainty about estimates for individual weights.

In-between uncertainty pertains to the uncertainty that exists in between separated regions of observations (Foong et al., 2019). We can consider this a type of epistemic uncertainty and can synthetically create datasets that include in-between uncertainty by removing data points in the middle of the data distribution.

Distributional uncertainty pertains to uncertainty that arises due to a mismatch between the training and test distributions and describes cases in which a model cannot confidently generalize to previously unseen test data (Malinin and Gales, 2018). Whereas Bayesian approaches often implicitly model distributional uncertainty as model uncertainty, Malinin and Gales (2018) proposes a framework called Prior Networks, which we briefly reviewed in §1.3, for distinctly handling distributional uncertainty. For the purposes of this thesis, however, we do not explicitly consider Prior Networks. Instead, we broadly treat distributional uncertainty as a form of epistemic uncertainty because it can be reduced with additional data and acknowledge that the implications for handling distributional uncertainty differ from those for handling model or parameter uncertainty. We also note that the concept of distributional uncertainty is closely related to out-of-distribution (OOD) detection, which we review as an approach for handling epistemic uncertainty in §3.3.3.

Approximation uncertainty arises from the inability of simple models to fit complex data and pertains to the difference between model approximations and true output values, as in the case of a basic linear regression (Tagasovska and Lopez-Paz, 2019). Tagasovska and

Lopez-Paz (2019) further note that because neural networks can be considered universal approximators (Cybenko, 1989), approximation uncertainty is negligible and can be omitted in the context of deep learning. We accept this notion and do not consider approximation uncertainty in this work.

3.2 Modeling aleatoric and epistemic uncertainty

The first two subsections of this section, §3.2.1 and §3.2.2, discuss approaches for modeling aleatoric and epistemic uncertainty separately that originate from Bayesian deep learning but can nevertheless be applied to traditional deep learning. The former applies to regression and the latter applies to both regression and classification.

The third subsection, §3.2.3, describes an approach proposed in Kendall and Gal (2017) that attempts to unify the approaches highlighted in §3.2.1 and §3.2.2 in a single framework. Kendall and Gal (2017) propose variants for both regression and classification.

Then, §3.2.4 discusses modeling aleatoric uncertainty with quantile regression, which is a more artisanal approach that differs from the preceding methods. We do not pair this method with any model nor metric for epistemic uncertainty; it remains distinct and provides a complementary perspective.

3.2.1 Aleatoric uncertainty as output variance in regression

The most common approach for modeling aleatoric uncertainty, as discussed in §1.3, is to assume that the conditional distribution of the target variable is Gaussian and approximate aleatoric uncertainty as noise that is learned by optimizing the per point model precision. This can be done in both traditional and Bayesian deep learning.

In traditional deep learning, we follow the approach mentioned in §2.1.1 and assign one output layer to estimate the variance of a Gaussian, such that $\mathbf{f}(\mathbf{x}) = (\hat{\mu}, \hat{\sigma}^2)$ is the output of the NN and Equation 2.4 is the variance-dependent training objective.

By comparison, in Bayesian deep learning, we use the standard BNN model outlined in Equation 2.5, where $\mathbf{y}_i \sim \mathcal{N}(\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i), \sigma_{\mathbf{y}}^2 \mathbf{I})$ and estimate the aleatoric uncertainity with the model precision

$$u_a(\mathbf{x}_i) = \hat{\sigma}_{\mathbf{y}}^2(\mathbf{x}_i) \tag{3.1}$$

However, one downside of this BNN approach is that it is equipped to model *homoscedastic* but not *heteroscedastic* variance, unlike the aforementioned traditional deep learning approach, in which the variance $\hat{\sigma}^2$ can be modeled as data-dependent.

Homoscedastic and *heteroscedastic* variance describe different assumptions about the nature of the noise in a dataset that is used for regression. Whereas homoscedastic variance assumes that the variance $\sigma^2_{\mathbf{y}}$ is identical for all inputs $\mathbf{x}$, heteroscedastic variance assumes that the variance $\sigma^2_{\mathbf{y}}(\mathbf{x}_i)$ varies with the input $\mathbf{x}_i$. Although homoscedasticity is a common assumption in many statistical models, it is important to design models that account for heteroscedastic noise when certain regions of the domain exhibit more inherent variance than others. Heteroscedasticity, in general, tends to be an important assumption for real-life datasets in which we do not know the underlying data-generating process. As such, it is important to use estimates of aleatoric uncertainty that model heteroscedasticity.

Modeling heteroscedastic variance in Bayesian neural networks

It is possible to adapt BNNs to model heteroscedastic aleatoric variance by constructing a network that is split to predict the mean $\mathbf{f}^{\mathbf{W}}(\mathbf{x})$ and the variance $\mathbf{g}^{\mathbf{W}}(\mathbf{x})$, where we also introduce some prior over the weights used for the variance. Then, the data-dependent likelihood can be expressed as $\mathbf{y}_i \sim \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$, where $\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i) = \hat{\mu}_i$ and $\mathbf{g}^{\mathbf{W}}(\mathbf{x}_i) = \hat{\sigma}_i^2$ (Gal, 2016). Naively speaking, we would then attempt to perform inference using one of the methods described in §2.2.3 to approximate the resulting posterior, which would be proportional to the likelihood and priors on the weights. The negative log of this posterior can be expressed as

$$-\log\left(p(\mathbf{W})\prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{x}_i,\mathbf{W})\right) = -\log p(\mathbf{W}) - \sum_{i=1}^{N}\log p(\mathbf{y}_i|\mathbf{x}_i,\mathbf{W}), \tag{3.2}$$

where the negative log likelihood (NLL) can be further simplified

$$\begin{aligned}-\sum_{i=1}^{N}\log p(\mathbf{y}_i|\mathbf{x}_i,\mathbf{W}) &= -\sum_{i=1}^{N}\log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_i^2}}\exp\left(-\frac{(\mathbf{y}_i-\hat{\mu}_i)^2}{2\hat{\sigma}_i^2}\right)\right)\\ &= \frac{N}{2}\log\left(2\pi\right) + \frac{1}{2}\sum_{i=1}^{N}\left(s_i + \exp(-s_i)\left(\mathbf{y}_i-\hat{\mu}_i\right)^2\right),\end{aligned} \tag{3.3}$$

where we set $s_i = \log\hat{\sigma}_i^2$ to be the NN's estimate for the variance for numerical stability.

While in theory it is possible to use HMC (§4.2.1) and BBB (§4.2.2) to approximate this posterior, in practice we find that neither method is able to produce an accurate posterior predictive. Empirically speaking, it is likely that minimization of the NLL in Equation 3.3 yields weights that produce abnormally large variance estimates s_i in order to minimize $\exp(-s_i)\left(\mathbf{y}_i-\hat{\mu}_i\right)^2$, in turn allowing the predictive mean μ_i to deviate wildly from the ground truth $\mathbf{y}_i$.

Thus, given the inability of HMC and BBB to accurately approximate the posterior of the heteroscedastic aleatoric variance model, we instead opt to use MC dropout (§2.2.3), as proposed by Gal (2016). We adapt the NLL in Equation 3.3 to get the tractable minimization objective

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \exp(-s_i) \left(\mathbf{y}_i - \hat{\mu}_i\right)^2 + \frac{1}{2} s_i, \tag{3.4}$$

which we can use as our loss function in MC dropout.

We then model aleatoric uncertainty as the heteroscedastic variance

$$u_a(\mathbf{x}_i) = \frac{1}{T} \sum_{t=1}^{T} \hat{\sigma}_{i,t}^2, \tag{3.5}$$

where we average over T forward passes through the network.

3.2.2 Epistemic uncertainty as variance due to model sampling

Because epistemic uncertainty is most commonly conceptualized in terms of model and parameter uncertainty (as discussed in §3.1), the most typical approach for obtaining estimates of epistemic uncertainty entails capturing the range of possible functions that a NN model approximates. This is usually done by sampling from the model to generate different function realizations, which can be visualized with a posterior predictive, and subsequently computing the variance in these realizations as an estimate for epistemic uncertainty (Gal, 2016).

For a NN that performs regression and produces an output $\mathbf{f}^{\mathbf{W}}(\mathbf{x_i})$ as an estimate for the output $\mathbf{y_i}$, the epistemic uncertainty for the input $\mathbf{x}_i$ can be expressed as the variance of the model output $\mathbf{f}^{\mathbf{W}}(\mathbf{x_i})$,

$$u_e(\mathbf{x}_i) = \frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{f}^{\mathbf{W}_t}(\mathbf{x}_i)\right)^2 - \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{f}^{\mathbf{W}_t}(\mathbf{x}_i)\right)^2, \tag{3.6}$$

where we stochastically sample from the model with T random passes (Gal, 2016). We note that $\frac{1}{T} \sum_{t=1}^{T} \mathbf{f}^{\mathbf{W}_t}(\mathbf{x}_i)$ is the predictive mean.

Although this is not proposed by Gal and Ghahramani (2016), we can also apply this estimate to binary classification tasks, where we still use the model output $\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i) = \hat{p}_i$ in Equation 3.6, noting that the only difference functionally is that $\hat{p}_i \in [0, 1]$.

This estimate can also be extrapolated to multiclass classification in a principled way. Considering that a NN used for classification outputs a C-dimensional probability vector $\mathbf{p}_i$ that contains the probabilities of an input being classified with a label in the set $\{1, ..., C\}$, we compute the variance of the neural network prediction $\hat{p}_{i,c}$ for each class c and average over all the classes, such that

$$u_e(\mathbf{x}_i) = \sum_{c=1}^{C} \left(\frac{1}{T} \sum_{t=1}^{T} (\hat{p}_{i,t,c})^2 - \left(\frac{1}{T} \sum_{t=1}^{T} (\hat{p}_{i,t,c}) \right)^2 \right), \tag{3.7}$$

where we are effectively we are averaging over the variance of C binomially distributed random variables $p_{i,c}$. Although this epistemic uncertainty estimate is based on Gal and Ghahramani (2016), it is to our knowledge unprecedented in the literature and, as such, a novel theoretical contribution.

Finally, we note that while the premise of estimating epistemic uncertainty based on differences in model sampling was originally applied to BNNs, it can also be applied to ensembles of deterministic neural networks. To use variance in model sampling as an estimate for epistemic uncertainty in an ensemble, we stochastically sample T predictions for $\mathbf{y}_i$ from the distribution produced by the ensemble.

3.2.3 Combining aleatoric and epistemic uncertainty in one model

Kendall and Gal (2017) propose a model that combines the two preceding approaches to model aleatoric uncertainty as heteroscedastic variance and epistemic uncertainty as variance that exists due to model sampling.

For regression, we use the network described in 3.2.1, which is split to predict the mean $\mathbf{f}^{\mathbf{W}}(\mathbf{x})$ and the variance $\mathbf{g}^{\mathbf{W}}(\mathbf{x})$. We place a distribution over the weights of this network and fix a Gaussian likelihood to model aleatoric uncertainty. Then, we use MC dropout to minimize the objective in Equation 3.4 and model aleatoric uncertainty with Equation 3.5 and epistemic uncertainty with Equation 3.6. We show that this is a generalization of the law of total variance in §3.3.2.

Heteroscedastic classification

It is possible to extend the approach described above to classification by marginalizing over intermediate heteroscedastic regression uncertainty placed over the logit space (Kendall and Gal, 2017). This produces a *heteroscedastic classification* model different from any model previously considered.

We set up a NN to output the C-dimensional vectors $\mathbf{f}_i^{\mathbf{W}}$ and $\mathbf{g}_i^{\mathbf{W}}$ corresponding respectively to each label in a set $\{1, ..., C\}$ for some input $\mathbf{x}_i$. Then, we place a Gaussian distribution over this vector, such that

$$\begin{aligned} \hat{\mathbf{z}}_i|\mathbf{W} &\sim \mathcal{N}\left(\mathbf{f}_i^{\mathbf{W}}, \left(\sigma_i^{\mathbf{W}}\right)^2\right) \\ \hat{\mathbf{p}}_i &= \sigma(\hat{\mathbf{z}}_i), \end{aligned} \tag{3.8}$$

where σ is the softmax function as defined in Equation 2.7 and $\left(\sigma_i^{\mathbf{W}}\right)^2$ is a diagonal matrix containing the entries of $\mathbf{g}_i^{\mathbf{W}}$. Then, we can interpret this representation to mean that the model output $\mathbf{f}_i^{\mathbf{W}}$ is corrupted with Gaussian noise with some variance $\left(\sigma_i^{\mathbf{W}}\right)^2$ at an "intermediate" stage before we apply the softmax function to obtain the final class prediction vector $\hat{\mathbf{p}}_i$.

We marginalize over the intermediate heteroscedastic uncertainty to obtain the expected log likelihood

$$\log \mathbb{E}_{\mathcal{N}(\mathbf{z}_i;\mathbf{f}_i^{\mathbf{W}},(\sigma_i^{\mathbf{W}})^2)}\left[\hat{\mathbf{p}}_{i,c}\right],$$

where c is the observed class for an input $\mathbf{x}_i$. We approximate this log likelihood with Monte Carlo integration by sampling from a normal distribution, yielding the loss function

$$\mathcal{L}(\mathbf{W},\mathbf{b}) = \sum_{i=1}^{N} \log\left(\frac{1}{T}\sum_{t=1}^{T} \exp\left(\hat{\mathbf{z}}_{i,t,c} - \log\sum_{c'}\exp\left(\hat{\mathbf{z}}_{i,t,c'}\right)\right)\right), \tag{3.9}$$

where

$$\hat{\mathbf{z}}_{i,t} = \mathbf{f}_i^{\mathbf{W}} + \sigma_i^{\mathbf{W}}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}\left(0,\mathbf{I}\right)$$

and $\hat{\mathbf{z}}_{i,t,c'}$ is the element corresponding to the c' class in the vector $\hat{\mathbf{z}}_{i,t}$.

We model epistemic uncertainty with Equation 3.6 and can model aleatoric uncertainty in a few different ways, firstly as proposed by Kendall and Gal (2017) and secondly as adapted by Kwon et al. (2018) and Shridhar, Laumann, and Liwicki (2018). We also note that Kwon et al. (2018) and Shridhar, Laumann, and Liwicki (2018) use the same approach as Kendall and Gal (2017) for modeling epistemic uncertainty.

First, Kendall and Gal (2017) models aleatoric uncertainty with Equation 3.5, using the intermediate heteroscedastic variance $\left(\sigma_i^{\mathbf{W}}\right)^2$. We note, however, because the aleatoric uncertainty is modeled at an "intermediate" layer, this model is uniquely different from the previously discussed approaches for regression that model hetereoscedastic aleatoric variance at the final output. Given this unique model dependence for aleatoric uncertainty and lack of tractable analytical representations that can be used with other Bayesian approximate inference methods, we do not pair the heteroscedastic classification model with any methods that we describe in §4 besides Monte Carlo dropout. Instead, we evaluate this as its own end-to-end approach for estimating and quantifying aleatoric and epistemic uncertainty in order to examine the implications of using this intermediate layer representation.

Kwon et al. (2018) similarly note that this approach is deficient because it models the variance of the intermediate linear predictor $\hat{z}_i$ rather than the predictive probability $\hat{\mathbf{p}}_i$. Given this, they claim that this approach does not consider that the covariance matrix of the multinomial random variable $\left(\sigma_i^{\mathbf{W}}\right)^2$ is a function of the mean vector $\mathbf{f}_i^{\mathbf{W}}$ and that the estimate for aleatoric uncertainty does not account for correlations because the matrix is diagonal. Instead, they model aleatoric uncertainty at the output layer with

$$u_a(\mathbf{x}_i) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{diag}(\hat{\mathbf{p}}_t) - \hat{\mathbf{p}}_t\hat{\mathbf{p}}_t^{\mathrm{T}}, \tag{3.10}$$

which we note is equivalent to computing the variance for the Bernoulli random variable $\mathbf{p}_i, c$ for each category c.

Shridhar, Laumann, and Liwicki (2018) similarly use Equation 3.10 to model aleatoric uncertainty, but replace the softmax function used in Equation 3.8 with a normalized softplus function, such that the entries in the vector $\hat{\mathbf{p}}_i$ are instead computed with

$$\hat{\mathbf{p}}_{i,c} = \frac{\log\left(1 + \exp(\hat{\mathbf{z}}_{i,c})\right)}{\sum_{c'} \log\left(1 + \exp(\hat{\mathbf{z}}_{i,c'})\right)}.$$

Shridhar, Laumann, and Liwicki (2018) note that their rationale for using a normalized softplus function is that the normalized softplus function more easily produces vectors that are in practice zero, which in theory improves approximation for classification tasks. Whereas the softmax function rarely outputs predictions of zero because it requires an input logit of negative infinity to do so, the normalized softplus function only requires an input logit that is roughly smaller than -4 to output zero.

3.2.4 Aleatoric uncertainty using quantile regression

Quantile regression methods, similar to the Gaussian variance approach discussed in §3.2.1, are based upon the estimation of uncertainty in a conditional distribution of $\mathbf{Y}$. However, rather than modeling this conditional distribution using a Gaussian—which can only model symmetric and unimodal noise—quantile regression methods attempt to model a distribution function for $\mathbf{Y}$ using quantiles, which can more robustly describe the relationship between $\mathbf{X}$ and $\mathbf{Y}$ at different points in the conditional distribution and thus are capable of capturing asymmetry, multimodality, and heteroscedasticity.

A few different applications of quantile regression in deep learning attempt to model uncertainty in conditional distributions of the target variable (White, 1992; Taylor, 2000; Tagasovska and Lopez-Paz, 2019). Of these methods, we implement and evaluate the most modern and flexible model proposed by Tagasovska and Lopez-Paz (2019), called Simultaneous Quantile Regression (SQR). The remainder of this section reviews the technical details of SQR as outlined in Tagasovska and Lopez-Paz (2019).

Simultaneous Quantile Regression

Let $F(\mathbf{y}) = p(\mathbf{Y} \leq \mathbf{y})$ be the CDF of the target variable $\mathbf{Y}$ and $F^{-1}(\tau) = \inf\{\mathbf{y} : F(\mathbf{y}) \geq \tau\}$ be the quantile distribution function of $\mathbf{Y}$ for all quantile levels $0 \leq \tau \leq 1$. Then, the goal of quantile regression is to construct a model $\hat{\mathbf{y}} = \hat{f}_\tau(\mathbf{x})$ that approximates the conditional distribution function $\mathbf{y} = F^{-1}(\tau|\mathbf{X} = \mathbf{x})$.

The traditional approach to construct such a model is to use the pinball loss,

$$\ell_\tau(\mathbf{y}, \hat{\mathbf{y}}) = \begin{cases} \tau(\mathbf{y} - \hat{\mathbf{y}}), & \text{if } \mathbf{y} - \hat{\mathbf{y}} \geq 0, \\ (1-\tau)(\hat{\mathbf{y}} - \mathbf{y}), & \text{otherwise,} \end{cases}$$

which provides a theoretical basis for SQR.

Then, the optimal quantile distribution $\hat{f}_\tau$ that minimizes this loss can be computed with

$$\hat{f}_\tau \in \operatorname*{argmin}_f \frac{1}{N} \sum_{i=1}^{N} \ell_\tau \left(f(\mathbf{x}_i), \mathbf{y}_i \right). \tag{3.11}$$

Tagasovska and Lopez-Paz (2019) propose estimating all of the quantile levels τ for this loss function simultaneously with

$$\hat{f} \in \operatorname*{argmin}_f \frac{1}{N} \sum_{i=1}^{N} \mathop{\mathbb{E}}_{\tau \sim \mathrm{U}(0,1)} \left[\ell_\tau \left(f_\tau(\mathbf{x}_i), \mathbf{y}_i \right) \right],$$

which is solved in practice by sampling $\tau \sim \mathrm{U}(0,1)$ and using stochastic mini-batching. This resulting function $\hat{f}_\tau(\mathbf{x})$ models the conditional distribution of $\mathbf{y}$ given some input $\mathbf{x}$. Aleatoric uncertainty can then be modeled with a $1-\alpha$ prediction interval around the median estimate produced by $\hat{f}_\tau(\mathbf{x})$,

$$u_a(\mathbf{x}_i) = f\left(\mathbf{x}_i, 1 - \frac{\alpha}{2}\right) - f\left(\mathbf{x}_i, \frac{\alpha}{2}\right). \tag{3.12}$$

One SQR model constructed in this fashion captures the entire conditional distribution of $\mathbf{Y}$. Moreover, because no ensembling or approximate inference is required, SQR—which is deployed with a standard deterministic NN—is attractive from a computational standpoint.

3.3 Quantifying aleatoric and epistemic uncertainty

We consider three categories of approaches for quantifying uncertainty in this section.

The first is traditional measures of variability for categorical data, which generally produce estimates for total uncertainty. By coupling them with one of the decompositions described in §3.3.2, we show how many of these metrics can be used to arrive at separate estimates for aleatoric and epistemic uncertainty.

The second category is decompositions—i.e., ways to decompose total uncertainty into its aleatoric and epistemic components based on statistical theory and principles. We apply the first decomposition to the measures of variability for categorical data and demonstrate that the second decomposition is a generalization of previously discussed approaches in §3.2.

Finally, the third category is out-of-distribution (OOD) detection metrics, which can be considered as proxies for epistemic uncertainty.

3.3.1 Measures of variability for categorical data

Although literature from classical statistics and adjacent disciplines has proposed many different variability measures for summarizing categorical data, few measures have achieved widespread acceptance by practitioners (Allaj, 2018). We consider four measures in this thesis, all of which—by their nature—can be applied only in a classification setting.

The first two—variation ratio and predictive entropy—have been applied in a deep learning context (Gal, 2016). The latter two—the Gini-Simpson index and a closely related measure to which we will refer as the Allaj index (Allaj, 2018)—are functions of the relative frequencies of the different categories in a distribution and are not typically considered in deep learning. We explore application of these metrics in a classification setting in an effort to examine whether there are statistical alternatives to predictive entropy that exhibit different properties.

Variation ratio

Variation ratio is a measure of dispersion that is defined as the proportion of cases that are not in the modal category,

$$\mathbf{VR}[\mathbf{x}] = 1 - \frac{f_{\mathbf{x}}}{T}, \tag{3.13}$$

where $f_{\mathbf{x}}$ is the number of cases with the mode and T is the total number of cases (Freeman, 1965). In a deep learning context, we collect T stochastic samples from the distribution produced by a neural network and treat $f_x = \sum_t \mathbb{1}\left[\mathbf{y}_t = c^*\right]$ as the number of times the modal class c^* was sampled. We note that variation ratio has a minimum of 0 and maximum of 0.5 in the case of binary classification.

Predictive entropy

Predictive entropy is a metric that stems from information theory and attempts to capture the amount of information inherent in a predictive distribution (Shannon, 1948). It is defined as

$$\mathbf{H}[\mathbf{y}|\mathbf{x}, \mathbf{W}] = -\sum_{c=1}^{C} p(\mathbf{y} = c|\mathbf{x}, \mathbf{W}) \log p(\mathbf{y} = c|\mathbf{x}, \mathbf{W}) \tag{3.14}$$

and is maximized when all classes are equally probable and minimized with a value of 0 when one class has a probability of 1.

Gini-Simpson index

The Gini-Simpson index was originally proposed as a statistical formula (Gini, 1912) and later extended to ecology to be used as a measure of biodiversity (Simpson, 1949). We can define it in the context of deep learning as

$$\mathbf{GS}[\mathbf{y}|\mathbf{x},\mathbf{W}] = 1 - \sum_{c=1}^{C} p(\mathbf{y}=c|\mathbf{x},\mathbf{W})^2, \tag{3.15}$$

where the index is maximized and minimized similarly to predictive entropy.

Allaj index

We coin the term "Allaj index" to refer to the measure proposed in Allaj (2018), which is closely related to the Gini-Simpson index. We define it in the context of deep learning as

$$\mathbf{A}[\mathbf{y}|\mathbf{x},\mathbf{W}] = 1 - \left(\sum_{c=1}^{C} p(\mathbf{y}=c|\mathbf{x},\mathbf{W})^2\right)^{\frac{1}{2}}, \tag{3.16}$$

noting that it is also maximized and minimized similarly to predictive entropy.

3.3.2 Decompositions

Depeweg et al. (2017) propose the notion of decomposing uncertainty into its aleatoric and epistemic components in a principled way in the context of BNNs with latent input variables (BNN+LV), a family of models introduced in Depeweg et al. (2016) to describe complex stochastic patterns.

The first decomposition is based upon conditioning on a specific value of $\mathbf{W}$ to arrive at an estimate of mutual information from the predictive entropy (as defined in §3.3.1), which was first proposed in Houlsby et al. (2011) in the context of deep learning. The second deecomposition is based upon the law of total variance (LOTV).

Instead of relying on the BNN+LV model, we extend these decompositions to the more general context of NNs that allow us to model outputs with predictive distributions (i.e., either Bayesian or ensemble methods). For the first decomposition, we entertain the possibility of using the Gini-Simpson index and Allaj index as an alternative to the predictive entropy. For the second decomposition, we demonstrate that the law of total variance is a generalization of the approaches proposed by Gal (2016), Shridhar, Laumann, and Liwicki (2018) and Kwon et al. (2018) (as described in §3.2.3).

Conditioning and mutual information

The total uncertainty present in a probability distribution $p(y^*|x^*)$ can be quantified in terms of predictive entropy with $\mathbf{H}[y^*|x^*]$ (Depeweg et al., 2017). Then, in the context of a deep learning model that seeks to approximate $p(y^*|x^*)$ with some parameters $\mathbf{W}$, we can remove the parameter uncertainty by conditioning on a specific value of $\mathbf{W}$. We do this in a principled way by taking the expectation of the the conditional predictive entropy under $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$, producing

$$u_a(x^*) = \mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\, \mathbf{H}[y^*|x^*, \mathbf{W}] \tag{3.17}$$

as an estimate for the aleatoric uncertainty present in $p(y^*|x^*)$, as we have now removed the uncertainty about the parameters, which we think of as a source of epistemic uncertainty.

We can then quantify the epistemic uncertainty by taking the difference between the total and aleatoric uncertainty,

$$u_e(x^*) = \mathbf{H}[y^*|x^*] - \mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\, \mathbf{H}[y^*|x^*, \mathbf{W}], \tag{3.18}$$

which is also known as the *mutual information* between the prediction y^* and the posterior over the model parameters $\mathbf{W}$ (Houlsby et al., 2011). Mutual information is maximized on points for which the model is generally uncertain but for which there are still parameters $\mathbf{W}$ that produce incorrect predictions with a high level of confidence (Gal, 2016).

In this work, we propose a novel extension of the above decomposition—marginalizing over $\mathbf{W}$ to obtain an estimate for aleatoric uncertainty—to other measures of variability in categorical data. In principle, given that predictive entropy is a measure of dispersion that is structurally similar to the Gini-Simpson index and the Allaj index (§3.3.1), we can marginalize over $\mathbf{W}$ as described above to yield the aleatoric and epistemic components of these indices. For the Gini-Simpson index, we can express this with

$$u_a(x^*) = \mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\, \mathbf{GS}[y^*|x^*, \mathbf{W}] \tag{3.19}$$

$$u_e(x^*) = \mathbf{GS}[y^*|x^*] - \mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\, \mathbf{GS}[y^*|x^*, \mathbf{W}], \tag{3.20}$$

where total uncertainty is the sum of $u_a(x^*)$ and $u_e(x^*)$.

To obtain this decomposition for the Allaj index, we simply substitute $\mathbf{GS}[\,\cdot\,]$ with $\mathbf{A}[\,\cdot\,]$ in Equation 3.19. We note that we do not apply this decomposition to the variation ratio because it is a metric that is computed based upon proportion of samples not in the modal category and for which we therefore cannot condition upon $\mathbf{W}$.

Law of total variance

Depeweg et al. (2017) further proposes using the law of total variance (LOTV) to quantify the total uncertainty in a distribution. Letting $\sigma^2(\cdot)$ represent variance, we can decompose the total uncertainty in a probability distribution $p(y^*|x^*)$ with

$$\sigma^2(y^*|x^*) = \underbrace{\mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}[\sigma^2(y^*|x^*, \mathbf{W})]}_{u_a(x^*)} + \underbrace{\sigma^2_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\left(\mathbb{E}[y^*|x^*, \mathbf{W}]\right)}_{u_e(x^*)}. \tag{3.21}$$

The first term $\mathbb{E}_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}[\sigma^2(y^*|x^*, \mathbf{W})]$ represents the average value of $\sigma^2(y^*|x^*)$ over the posterior for $\mathbf{W}$. Because it ignores any contribution to the variance of $p(y^*|x^*)$ from $\mathbf{W}$ and continues to exist even as we obtain more data and the posterior for $\mathbf{W}$ concentrates, it represents the aleatoric uncertainty.

The second term $\sigma^2_{p(\mathbf{W}|\mathbf{X},\mathbf{Y})}\mathbb{E}[y^*|x^*, \mathbf{W}]$ represents the uncertainty that exists in $p(y^*|x^*)$ due to variance in $\mathbf{W}$—i.e., epistemic uncertainty. As we gather data and the posterior for $\mathbf{W}$ concentrates, this variance goes to 0, indicating that epistemic uncertainty disappears as desired.

Considering this decomposition more deeply, we realize that it is mathematically equivalent to the traditional ways of modeling aleatoric and epistemic uncertainty that are described in §3.2.1, §3.2.2, and, by extension, §3.2.3. Indeed, Equation 3.5, which computes the aleatoric uncertainty as the average heteroscedastic variance over T stochastic samples from the model, corresponds to the first term $u_a(x^*)$ in Equation 3.21 above. Similarly, Equation 3.6, which computes the epistemic uncertainty as the variance that exists due to stochastically sampling T different $\mathbf{W}$, corresponds to the second term $u_e(x^*)$ in Equation 3.21. For regression, we semantically refer to the approach described in these sections as the law of total variance (LOTV) as proposed by Kendall and Gal (2017).

For classification, we semantically refer to the three approaches used for the heteroscedastic classification model in §3.2.3 as the LOTV as proposed by Kendall and Gal (2017), Shridhar, Laumann, and Liwicki (2018), and Kwon et al. (2018) respectively.[1]

3.3.3 Out-of-distribution detection

Although the literature on out-of-distribution (OOD) detection is broad and draws from a number of disciplines, we focus our implementation on some of the most well known approaches in machine learning, noting that all of these metrics can be treated as estimates for epistemic uncertainty.

It is important to note that, in general, the literature on OOD detection is focused primarily on classification, whereby traditional classification datasets are split into "in-domain"

[1] Technically speaking, because Kendall and Gal (2017) model aleatoric uncertainty as heteroscedastic variance at an interemediate layer rather than the output layer in the heteroscedastic classification model, their proposed approach for classification is not formally a generalization of the LOTV (which computes the variance of the true output). However, we refer to it as such for ease of reference.

and "out-of-domain" classes. Far fewer papers, by comparison, propose methods for OOD detection for regression, which by definition lacks a formal way to define what is or is not out-of-domain.

While some novel literature has attempted to model in-distribution features in regression with generative models (such as simple Gaussian mixture models) and claim to achieve state-of-the-art OOD detection results, they are not well documented or reviewed. For this reason, we do not consider them in the scope of this work.

Instead, we consider the following OOD detection metrics, the first five of which are model-agnostic metrics that are simple to implement on top of existing architectures. The first (real distance) is exclusive to regression. The second (last layer distance) can be used for both regression and classification. The third, fourth, and fifth (largest softmax score, functional margin, and Gaussian) are exclusive to classification. Finally, the last approach, Orthonormal Certificates, is a more computationally costly, comparatively artisanal approach that can be applied to both regression and classification but that we evaluate only for regression in this work.

Real distance

We compute the smallest distance from an input $\mathbf{x}_i$ to the points in the training data $\mathbf{X}_{\text{train}}$. Larger values are interpreted as reflect greater uncertainty. While it is possible to use any norm in theory, we evaluate only the Euclidean norm in this work.

Last layer distance

We compute the smallest distance from the NN's last layer representation of an input $\mathbf{x}_i$ to the last layer representation of the points in the training data $\mathbf{X}_{\text{train}}$, where larger values reflect greater uncertainty. We again limit ourselves to the Euclidean norm.

Largest softmax score

We take the largest logit prediction for a given class for an input $\mathbf{x}_i$ and treat it as a proxy for uncertainty, where (unlike our preceding two metrics) larger values indicate less uncertainty.

Functional margin

We compute the difference between the two largest logit predictions for an input $\mathbf{x}_i$ and treat it as a proxy for uncertainty. Similar to largest softmax score, larger values indicate less uncertainty. We note that for the case of binary classification, this metric produces a very similar representation of uncertainty as largest softmax score.

Gaussian covariance

We fit a Gaussian to the NN's last layer representation of an input $\mathbf{x}_i$ and use its covariance as a proxy for epistemic uncertainty, where greater values indicate more uncertainty.

Orthonormal Certificates

Orthonormal Certificates (OCs), as proposed by Tagasovska and Lopez-Paz (2019), are a collection of diverse, non-constant functions that attempt to map training samples to zero and OOD examples to non-zero values. Per this approach, larger values reflect greater epistemic uncertainty. We review the technical details of OC as outlined in Tagasovska and Lopez-Paz (2019).

Let $\Phi = \{\phi(\mathbf{x}_i)\}_{i=1}^{N}$ be a high-level representation of training examples. Then, we can train a collection of certificates $C = (C_1, \ldots, C_K)$, where each certificate C_j is a simple NN that is trained to map the training dataset Φ to zero by minimizing some loss ℓ_c (which can be a typical loss function such as MSE). Then, epistemic uncertainty can be defined as

$$u_e(\mathbf{x}_i) = \left\| C^{\mathrm{T}} \phi(\mathbf{x}_i) \right\|^2, \tag{3.22}$$

which should evaluate to zero near the training distribution and to high values for inputs far from the training distribution.

We can implement k certificates on top of an h-dimensional representation of the training examples $\mathbf{x}_i$ as a single $h \times k$ layer, which would output a k-dimensional vector when paired with some loss ℓ_c. We note that the loss function ℓ_c is generally specified to be the same one that is used in the learning task.

Then, our OCs can be constructed with

$$\hat{C} \in \operatorname*{argmin}_{C \in \mathbb{R}^{h \times k}} \frac{1}{N} \sum_{i=1}^{N} \ell_c \left(C^{\mathrm{T}} \phi(\mathbf{x}_i), 0 \right) + \lambda \left\| C^{\mathrm{T}} C - \mathbf{I}_k \right\|, \tag{3.23}$$

where λ is a rate set to impose an orthonormality constraint between certificates so that the certificates are diverse and non-constant.

While OCs can be applied to both regression and classification, we apply them only to regression in this thesis and use the previously discussed distance-based estimators of epistemic uncertainty for classification. Tagasovska and Lopez-Paz (2019) note that when MSE is selected as the loss ℓ_c, the OCs seek the the directions in the data with the lowest variance and, by extension, estimate the least-variant components of the training features. This can be interpreted in terms of Principal Component Analysis (PCA) as corresponding to the principle components that are associated with the smallest singular values of the training features.

Chapter Four

Models and Inference Methods

In recent years, the field of deep learning has witnessed an abundance of novel proposals for scalable and practical inference methods that can be used to approximate models and obtain estimates of predictive uncertainty (with many of the approaches for Bayesian NNs reviewed in §2.2.3). While the rapid increase in the number of inference methods is at once exciting and promising for the field, there is limited existing literature focused on directly evaluating and comparing the uncertainty estimates produced for models that are approximated using these methods. Yao et al. (2019), for one, computes the quality of predictive uncertainty estimates in terms of commonly used metrics (MSE, NLL, PICP, MPIW) for models approximated with ten of the most common inference methods, but there is no existing work evaluating how other uncertainty metrics are affected by selection of inference methods, at least at the time that this thesis was written.

As such, as mentioned in the introduction, although the overarching goal of this work is primarily to evaluate different metrics for aleatoric and epistemic uncertainty, one of the supporting objectives is to assess the quality of these metrics when coupled with different inference methods. This, in turn, will provide practical knowledge as to which end-to-end frameworks (meaning model, method, *and* metric) for quantifying aleatoric or epistemic uncertainty are most useful.

We consider seven different approaches, which we can categorize under the broad philosophical umbrella of frequentist or Bayesian approaches.

1. The first model is a basic deterministic NN. For regression, the NN can output a pointwise prediction $\hat{y}$ or, alternatively, output the mean and variance of a normal distribution $(\hat{\mu}, \hat{\sigma}^2)$, as described in §2.1.1; we refer to this latter type of NN semantically as a "probabilistic NN" for the remainder of this thesis. For classification, the NN outputs a pointwise sigmoid probability $p \in (0, 1)$; we note that we can treat this as a proxy for aleatoric uncertainty, simply assuming that a model is more confident the closer p is to 0 or 1. Because we reviewed traditional NNs—which are neither frequentist nor Bayesian, strictly speaking—in §2.1, we do not discuss them in this

chapter.

2. The next two models are two variants of ensemble approaches—traditional ensembles and deep ensembles (Lakshminarayanan, Pritzel, and Blundell, 2017)—which are representative of a frequentist view of uncertainty.

3. The next three—Hamiltonian Monte Carlo (HMC), Bayes by Backprop (BBB), and Monte Carlo (MC) dropout—are inference methods that correspond to the three categories of Bayesian approximate inference for BNNs reviewed in §2.2.3.

4. Finally, the last model, known as the neural linear (NL) model, probabilistically models the weights only in the last layer of the NN and otherwise handles the preceding weights as in a traditional NN, thereby treating the rest of the network as a feature extractor. This model can also be regarded as applying Bayesian linear regression to the last layer.

We do not use all seven approaches for all of our experiments. Some, such as the deterministic NNs, are insufficiently expressive to distinguish aleatoric and epistemic uncertainty, and others, such as HMC and BBB, do not scale to large datasets. Note that two of the models—deep ensemble and the neural linear model—can only be applied to regression tasks.

4.1 Ensembles

Ensemble methods are based upon the premise of training many neural networks on the same data, relying on the stochasticity inherent in weight initialization and the training process to produce variance in the models' predictions. While the traditional ensemble approach models a distribution by combining the deterministic, pointwise outputs of many differently configured NNs (Hansen and Salamon, 1990), a number of more recent works have proposed averaging over the outputs of NNs that estimate the mean and variance of a normal distribution, as described in 2.1.1 (Osband, Blundell, et al., 2016; Lakshminarayanan, Pritzel, and Blundell, 2017; Tagasovska and Lopez-Paz, 2018). We implement and discuss both approaches here, referring to the first as *traditional ensembles* and to the second as *deep ensembles*.

4.1.1 Traditional ensembles

Traditional ensembles are simple to implement: we train M networks, each of which produces pointwise estimates for $\hat{\mathbf{y}}$, on the same data (which is randomly sampled or bootstrapped), randomly initializing weights for each of them. The collection of estimates is treated as a distribution and we compute the mean by averaging over the M estimates uniformly; this is also referred to as bootstrap aggregation or *bagging*.

While traditional ensembles are straightforward and practical, Yao et al. (2019) note that they fail to produce desirable uncertainty estimates because they rely on model diversity. In other words, because training objectives for ensemble methods do not inherently incorporate model diversity, it is possible that multiple models find similar local optima and resultantly produce poor uncertainty estimates. Problems with initialization may further contribute to this shortcoming.

4.1.2 Deep ensembles

Deep ensembles, also known as probabilistic ensembles, use the same approach as that of traditional ensembles but instead average over NNs that model normal distributions (which were described in §2.1.1. The ensemble is treated as a uniformly-weighted mixture model in which the predictions are combined per

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} p_{\theta_m}(\mathbf{y}|\mathbf{x}, \theta_m),$$

where p_{θ_m} and θ_m correspond to the output of the m-th model (Lakshminarayanan, Pritzel, and Blundell, 2017). As such, deep ensembles for classification simply average over the predicted probabilities and thus do not differ from the previously described traditional ensembles. By comparison, a deep ensemble for regression, in which we assume the use of a normal for the likelihood, produces a mixture of normal distributions

$$\frac{1}{M}\sum_{i=1}^{M} \mathcal{N}\left(\mu_{\theta_m}(\mathbf{x}), \sigma^2_{\theta_m}(\mathbf{x})\right),$$

for which the mean and variance of the mixture are given by

$$\mu(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} \mu_{\theta_m}(\mathbf{x})$$
$$\sigma^2(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} \left(\sigma^2_{\theta_m}(\mathbf{x}) + \mu^2_{\theta_m}(\mathbf{x})\right) - \mu^2(\mathbf{x})$$

respectively (Lakshminarayanan, Pritzel, and Blundell, 2017). Given this, we implement deep ensembles only for regression and not classification.

Lakshminarayanan, Pritzel, and Blundell (2017) argue that this method produces well-calibrated uncertainty estimates that are on par with BNNs, but note that there is further potential for improvement by de-correlating networks' predictions or optimizing ensemble weights. Given that this method presumably captures both aleatoric and epistemic uncertainty when paired with the law of total variance decomposition discussed in §3.3.2, we include it for evaluation and comparison with the others.

4.2 Bayesian models and inference methods

4.2.1 Hamiltonian Monte Carlo

We review MCMC methods as a class of approximate inference techniques (which include HMC) for BNNs in §2.2.3. Here, we review the technical details of HMC as developed by Neal et al. (2011).

Let $p(\theta)$ be a target posterior distribution with $\theta \in \mathbb{R}^D$. Then, at a high level, the goal of HMC is to use Hamiltonian dynamics to sample from $p(\theta)$ for parameters θ. In order to do this, we introduce an auxiliary momentum variable ρ, which is typically selected to be a multivariate normal that is independent of θ, and sample from the joint distribution $p(\rho, \theta) = p(\rho|\theta)p(\theta)$.

The physical properties of this system can be described with a function known as the Hamiltonian,

$$\begin{aligned} H(\rho, \theta) &= -\log p(\rho, \theta) \\ &= -\log p(\rho|\theta) - \log p(\theta) \\ &= K(\rho|\theta) + U(\theta), \end{aligned}$$

where $K(\rho|\theta) = -\log p(\rho|\theta)$ is the kinetic energy function and $U(\theta) = -\log p(\theta)$ is the potential energy function. Then, how this system changes over time is described by the partial derivatives of H:

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial \rho} = \frac{\partial K}{\partial \rho}, \qquad \frac{d\rho}{dt} = -\frac{\partial H}{\partial \theta} = -\frac{\partial U}{\partial \theta}$$

We approximate the dynamics represented by this differential equation by using the *leap-frog integrator*, a numerical integration algorithm that is reversible and volume-preserving, two special properties of Hamiltonian dynamics. The leapfrog integrator takes discrete time steps of some specified size ϵ. For each step, it samples a random momentum from $K(\rho|\theta) = K(\rho) \sim \mathcal{N}(0, M)$, where $M = m\mathbf{I}_{D\times D}$ is a matrix representing the mass of the system. Then, the leap-frog integrator simulates Hamiltonian motion with L leap-frog steps for each time step ϵ by iteratively updating the momentum with half-steps and the position with full steps:

$$\begin{aligned}
\rho &\leftarrow \rho - \frac{\epsilon}{2}\frac{\partial U}{\partial \theta} \\
\theta &\leftarrow \theta + \epsilon \frac{1}{m}\rho \\
\rho &\leftarrow \rho - \frac{\epsilon}{2}\frac{\partial U}{\partial \theta}
\end{aligned}$$

Because the approximation of the leap-frog numerical integrator is inexact, we apply a Metropolis-Hastings acceptance step to correct for simulation error. The probability α of accepting a new sample (ρ^*, θ^*) generated from an old sample (ρ, θ) is computed as

$$\alpha = \min\left(1, \exp\{H(\rho, \theta) - H(\rho^*, \theta^*)\}\right),$$

where the old sample is used again if the new sample is not accepted.

The HMC algorithm begins with some random position $\theta^{(0)}$ and proceed to sample $\theta^{(k)}$ for a given number of iterations k according to the process specified above. In short, the position θ is updated with some randomly sampled momentum ρ according to the properties of Hamiltonian dynamics, which are approximated numerically with a leap-frog integrator.

4.2.2 Bayes by Backprop

Bayes by Backprop (BBB) (Blundell et al., 2015) and other variational methods, as reviewed in §2.2.3, seek to approximate the target posterior distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ ofo a BNN with a variational distribution $q_\theta(\mathbf{W})$, where θ represents the parameters we wish to optimize.

The core of variational approaches is therefore to find some θ^* that minimizes the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) between $q_\theta(\mathbf{W})$ and $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$, i.e., such that

$$\theta^* = \operatorname*{argmin}_\theta \mathrm{KL}\big[q_\theta(\mathbf{W}) \,||\, p(\mathbf{W}|\mathbf{X}, \mathbf{Y})\big],$$

where KL divergence, also known as relative entropy, is defined as

$$\mathrm{KL}\big[q_\theta(\mathbf{W}) \,||\, p(\mathbf{W}|\mathbf{X}, \mathbf{Y})\big] = \int q_\theta(\mathbf{W}) \log \frac{q_\theta(\mathbf{W})}{p(\mathbf{W}|\mathbf{X}, \mathbf{Y})}\, d\mathbf{W}$$

and, since the above integral is intractable and must be approximated using repeated sampling, is often expressed in terms of expectation:

$$\mathrm{KL}\big[q_\theta(\mathbf{W}) \,||\, p(\mathbf{W}|\mathbf{X}, \mathbf{Y})\big] = \mathbb{E}_{q_\theta(\mathbf{W})}\left[\frac{q_\theta(\mathbf{W})}{p(\mathbf{W}|\mathbf{X}, \mathbf{Y})}\right]. \tag{4.1}$$

We note that minimizing the KL divergence is equivalent to maximizing the *evidence lower bound* (ELBO), or *variational lower bound*, subject to a KL complexity term on the parameters of the network. This yields the variational objective

$$\mathcal{L}(\theta) = \mathbb{E}_{q_\theta(\mathbf{W})}\big[\log p(\mathbf{Y}|\mathbf{X},\mathbf{W})\big] - \mathrm{KL}\big[q_\theta(\mathbf{W})\,||\,p(\theta)\big], \tag{4.2}$$

where the first term ensures that the variational distribution $q_\theta(\mathbf{W})$ represents the data well and the second term acts as a penalty that limits the extent to which $q_\theta(\mathbf{W})$ can deviate from the prior $p(\mathbf{W})$.

In order to find the optimal variational parameters, it is necessary to take the gradient of the ELBO with respect to θ. This, however, is not trivial to compute, as the gradient cannot be pushed into the expectation, which is also taken with respect to θ. As such, in order to perform this computation, we use the reparametrization trick proposed by Kingma and Welling (2013) and perform BBB.

Let us assume that we are using the mean-field Gaussian variational family, such that $q(\mathbf{W}|\mu,\mathbf{\Sigma}) = \mathcal{N}(\mathbf{W};\mu,\mathbf{\Sigma})$. Then, instead of sampling $W \sim q(\mathbf{W}|\mu,\Sigma)$, we can sample $\epsilon \sim \mathcal{N}(0,\mathbf{I})$ and set the network parameters to $\mathbf{W} = \mu + \epsilon^{\mathrm{T}}\mathbf{\Sigma}^{1/2}$, where $\mathbf{I}$ and $\mathbf{\Sigma}$ have the same dimensions. We then perform backpropagation ass normal and take the gradients of $f(\epsilon) = \mu + \epsilon^{\mathrm{T}}\mathbf{\Sigma}^{1/2}$ with respect to $\mathbf{W},\mu,\mathbf{\Sigma}$. Finally, we can update our parameters $\mu,\mathbf{\Sigma}$ according to these gradients,

$$\mu \leftarrow \mu - \eta\left(\frac{\partial f}{\partial \mathbf{W}} + \frac{\partial f}{\partial \mu}\right)$$
$$\mathbf{\Sigma} \leftarrow \mathbf{\Sigma} - \eta\left(\frac{\partial f}{\partial \mathbf{W}}\frac{\epsilon}{\mathbf{\Sigma}} + \frac{\partial f}{\partial \mathbf{\Sigma}}\right),$$

where η is the learning rate. Practically speaking, we initialize with $\mu^{(0)},\mathbf{\Sigma}^{(0)}$ and iterate BBB for a given number of iterations k to achieve parameters closer to the optimal $\mu^*,\mathbf{\Sigma}^*$ that maximize the ELBO.

4.2.3 Monte Carlo dropout

Per Monte Carlo (MC) dropout, as described in 2.2.3 and proposed by Gal and Ghahramani (2016), we train a traditional NN with dropout, where individual units in a layer i are retained with some probability p_i, effectively applying an independent random Bernoulli mask to the units. Then, instead of turning off dropout for inference, we apply it as we sample from the NN with T stochastic forward passes. The resulting distribution of the outputs from these T passes is the posterior predictive, from which it becomes possible to compute uncertainty estimates.

Gal and Ghahramani (2016) further note that because the dropout objective minimizes the KL divergence between an approximating distribution and the posterior of a deep Gaussian process, a NN with dropout applied before every layer is mathematically equivalent to an approximation of a deep Gaussian process (Damianou and Lawrence, 2013). It is possible to perform moment-matching and estimate the first two moments of the predictive distribution empirically with the sample mean and variance:

$$\frac{1}{T}\sum_{i=1}^{T} \mathbf{f}^{\hat{\mathbf{W}}_t}(\mathbf{x}^*) \xrightarrow{T\to\infty} \mathbb{E}_{q_\theta(\mathbf{W})}[\mathbf{y}^*]$$

$$\tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{i=1}^{T} \mathbf{f}^{\hat{\mathbf{W}}_t}(\mathbf{x}^*)^{\mathrm{T}}\mathbf{f}^{\hat{\mathbf{W}}_t}(\mathbf{x}^*) - \tilde{\mathbb{E}}[\mathbf{y}^*]^{\mathrm{T}}\tilde{\mathbb{E}}[\mathbf{y}^*] \xrightarrow{T\to\infty} \mathrm{Var}_{q_\theta(\mathbf{W})}[\mathbf{y}^*]$$

where $\mathbf{f}^{\mathbf{W}}(\mathbf{x}^*)$ is the NN output for some input $\mathbf{x}^*$, $p(\mathbf{y}^*|\mathbf{f}^{\mathbf{W}}(\mathbf{x}^*)) = \mathcal{N}(\mathbf{y}^*; \mathbf{f}^{\mathbf{W}}(\mathbf{x}^*), \tau^{-1}\mathbf{I})$, τ is the model precision, $\hat{\mathbf{W}}_t \sim q_\theta(\mathbf{W})$, and $q_\theta(\mathbf{W})$ is the approximate distribution.

We note that this particular version of MC dropout described here assumes homoscedastic variance, but MC dropout can be extended to incorporate heteroscedasticity as proposed by Kendall and Gal (2017) and discussed in §3.2.1.

4.2.4 Neural linear model

The neural linear (NL) model, in which we perform exact inference on the last layer of weights in a NN and treat the remainder of weights as hyperparameters, is a more tractable alternative to BNNs that has been used in Bayesian optimization, active learning, and reinforcement learning (Snoek et al., 2015; Riquelme, Tucker, and Snoek, 2018; Pinsler et al., 2019; Ober and Rasmussen, 2019). In general, the NL model can be thought of as an approximation for a full BNN that is computationally advantageous and competitive in performance with BBB and MC dropout but that requires substantial hyperparameter tuning (Ober and Rasmussen, 2019).

Given a set of N observations of training data $\{\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}\} = \{(\mathbf{x}_1, \mathrm{y}_1),..., (\mathbf{x}_N, \mathrm{y}_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}$, let the outputs of the last hidden layer of the NN, which are parameterized by all the preceding weights θ, be represented with the matrix

$$\Phi_\theta = [\phi_\theta(\mathbf{x}_1), \ldots, \phi_\theta(\mathbf{x}_N)]^T.$$

Then, our model is defined as

$$\mathbf{Y} = \Phi_\theta \mathbf{W} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{N+1}),$$

where $\mathbf{W}$ includes a bias term and each $\phi_\theta(\mathbf{x})$ is augmented with a one to account for this term. We use a normal prior for the weights $p(\mathbf{W}) \sim \mathcal{N}(0, \mathbf{V}_0)$. Then, it follows that the posterior can be expressed as

$$\begin{aligned} p(\mathbf{W}|\mathbf{X}, \mathbf{Y}, \sigma^2) &= \mathcal{N}(\mathbf{W}|\mathbf{W}_N, \mathbf{V}_N) \\ &\propto \mathcal{N}(\mathbf{W}; 0, \mathbf{V}_0)\, \mathcal{N}(\mathbf{Y}; \Phi_\theta \mathbf{W}, \sigma^2 \mathbf{I}_{N+1}), \end{aligned}$$

where the mean and variance respectively are given by

$$\begin{aligned} \mathbf{W}_N &= \frac{1}{\sigma^2} \mathbf{V}_N \Phi_\theta^{\mathrm{T}} \mathbf{Y} \\ \mathbf{V}_N &= \left(\mathbf{V}_0^{-1} + \frac{1}{\sigma^2} \Phi_\theta^{\mathrm{T}} \Phi_\theta \right)^{-1}. \end{aligned}$$

The posterior predictive distribution for an input x^* therefore is

$$p(y^*|x^*, \mathbf{X}, \mathbf{Y}, \sigma^2) = \mathcal{N}(y^*; \mathbf{W}_N^{\mathrm{T}} \phi_\theta(x^*), \sigma^2 + \phi_\theta(x^*)^{\mathrm{T}} \mathbf{V}_N \phi_\theta(x^*)).$$

There are a number of ways to learn the hyperparameters θ. One way to do so is by following the approach proposed by Snoek et al. (2015), in which we set θ to the maximum a posterior (MAP) estimates for the corresponding weights and biases of a NN that is trained to maximize the objective

$$\mathcal{L}(\theta_{\text{full}}) = \mathcal{N}(\mathbf{Y}; \Phi_\theta \mathbf{W}, \sigma^2) - \lambda ||\theta_{\text{full}}||^2,$$

where θ_{full} represents the parameters of all weights and biases in the NN, including those in the output layer, and λ is a regularization rate. Then, once we have set θ, we simply apply Bayesian linear regression as described above on the weights and bias in the last layer of the NN.

An alternative the MAP NL model is the regularized NL model, where the features are learned by optimizing the marginal likelihood with respect to the network weights preceding the output layer; in other words, the weights are treated and optimized as hyperparameters (Ober and Rasmussen, 2019).

Chapter Five

Evaluation on Synthetic Data

In this section, we perform experiments on one-dimensional regression and two-dimensional binary classification tasks so that it is possible to visualize the ground truth distributions. We consider three synthetic datasets for regression and classification, which are purposefully constructed so that we can evaluate whether our methods and metrics accurately capture the aleatoric and epistemic uncertainty in the data distributions.

We note that the methods and uncertainty metrics used differ for regression and classification, given that few of the uncertainty metrics discussed in §3 can be applied to regression. We consider seven of the methods discussed in §4 for regression and five for classification, and highlight the specific uncertainty metrics that we use for regression and classification in §5.1.1 and §5.2.1 respectively.

5.1 Regression

5.1.1 Experimental setup

Given a set of N observations of inputs and outputs $\{(\mathbf{x}_1, \mathrm{y}_1),..., (\mathbf{x}_N, \mathrm{y}_N)\}$ where $\mathbf{x}_i \in \mathbb{R}^1$ and $\mathbf{y}_i \in \mathbb{R}^1$, we consider two different models.

The first is a homoscedastic noise model (as introduced in §2.1). It assumes a likelihood of the form $\mathbf{y} = \mathbf{f}^{\mathbf{W}}(\mathbf{x}) + \epsilon$, where $\mathbf{f}^{\mathbf{W}}$ is a neural network that is parameterized with the weights $\mathbf{W}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the homoscedastic noise (and is also known as the model precision).

The second is a heteroscedastic noise model (as introduced in §3.2.1). It assumes a likelihood of the form $\mathbf{y}_i = \mathcal{N}(\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i))$, where $\sigma^2(\mathbf{x}_i)$ represents the data-dependent variance or, i.e., the heteroscedastic noise. Our neural network in this model is split to predict the mean $\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i)$ and the variance $\mathbf{g}^{\mathbf{W}}(\mathbf{x}_i) = \sigma^2(\mathbf{x}_i)$.

We acknowledge that model selection is a separate issue from method selection and—as argued by Yao et al. (2019)—note that issues associated with the approximation gaps arising from inference should be addressed separately from model selection. As such, it is important to consider that the heteroscedastic noise model is more expressive than the homoscedastic noise model, particularly with regard to its ability to capture variation in the level of noise across different regions of the data domain. Given that we are practically motivated to achieve the greatest level of expressiveness possible in order to best capture aleatoric and epistemic uncertainty, we choose to use the heteroscedastic noise model when possible, unless otherwise constrained by our selection of inference method.

Methods

We pair seven methods (i.e., all of the methods described in §4) with the two models above.

Given that we established in §3.2.1 that HMC and BBB are not well-suited to approximate the posterior of the heteroscedastic noise model and that the NL model is implicitly set up to model homoscedastic noise, we apply these inference methods to the homoscedastic noise model. We use MC dropout for both the homoscedastic and heteroscedastic noise models.

We use a single probabilistic NN, which models data-dependent variance, for the heteroscedastic noise model. Finally, we note that neither ensembles nor deep ensembles—unlike Bayesian methods—have theoretical guarantees as to how they model noise, although deep ensembles average over many probabilistic NNs modeling heteroscedastic noise.

Uncertainty metrics

We can compute uncertainty estimates in several different ways for regression. Some of these estimates arise as a consequence of modeling assumptions (§3.2) and others arise when we explicitly apply metrics for quantifying uncertainty (§3.3).

First, we can use the variance that results from model sampling for each method as a proxy for uncertainty. Although we might interpret this variance as corresponding to epistemic uncertainty per previous discussion (§3.2.2), this variance is typically visualized with the homoscedastic or heteroscedastic aleatoric noise as part of the posterior predictive. As such, the variance in the first column of visuals in Figures 5.1, 5.2, and 5.3 in the results, denoted with '+/- 2 std' in the legend, corresponds to the total predictive uncertainty in the absence of any decompositions. We note that computing variance from model sampling is possible for every method except for the probabilistic NN, which is a single-model method from which sampling is, by definition, not possible. In place of this, however, we use the probabilistic NN's estimates for the heteroscedastic noise, although it is technically a model for aleatoric uncertainty (§3.2.1), similarly denoting it with '+/- 2 std'. Then, the variance in the posterior predictive for each method corresponds primarily to the uncertainty arising from model sampling for all but the probabilistic NN. We plot this variance against the

x-inputs in the second column of images in Figures 5.1, 5.2, and 5.3.

Second, we can use two OOD detection metrics (§3.3.3), the real distance to the training data and the last layer distance to the training data, as proxies for epistemic uncertainty. We visualize these metrics for all seven methods in the third column of visuals in Figures 5.1, 5.2, and 5.3.

Third, we use the LOTV decomposition as proposed by Kendall and Gal (2017) (Equation 3.21) to arrive at separate estimates for aleatoric and epistemic uncertainty from the heteroscedastic noise model. In light of the fact that our other Bayesian inference methods are not equipped to approximate the heteroscedastic noise model, as discussed earlier, we use only MC dropout and deep ensembles for this decomposition.

Fourth, we use the two artisanal approaches proposed by Tagasovska and Lopez-Paz (2019), SQR (§3.2.4) and OCs (§3.3.3), as ways of obtaining estimates for aleatoric and epistemic uncertainty respectively. We plot the results from these methods alongside the LOTV decomposition in Figures 5.4, 5.5, and 5.6 so that we can directly compare these explicit representations for aleatoric and epistemic uncertainty.

Datasets

We use three synthetic datasets used in prior literature to encourage continuity of research and allow for direct comparison with prior work. Each of the datasets is designed to highlight different types of uncertainty.

Dataset 1 is from Yao et al. (2019). It is generated with

$$\mathbf{y} = 0.1\mathbf{x}^2 + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.25),$$

by uniformly sampling 80 training inputs and 20 validation inputs from $[-4,-1] \cup [1,4]$. 200 test inputs are uniformly sampled from $[-6,6]$. The gap in $(-1,1)$ reflects a region of in-between uncertainty. As such, the dataset is designed to highlight epistemic uncertainty with homoscedastic aleatoric uncertainty.

Dataset 2 is from Depeweg et al. (2017). It is generated with

$$\mathbf{y} = 7\sin(\mathbf{x}) + 3\left|\cos\left(\frac{\mathbf{x}}{2}\right)\right|\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1),$$

by sampling 600 training inputs and 150 validation inputs from three Gaussians with mean parameters $\{\mu_1 = -4, \mu_2 = 0, \mu_3 = 4\}$ and variance parameters $\{\sigma_1 = \frac{2}{5}, \sigma_2 = 0.9, \sigma_3 = \frac{2}{5}\}$, with each Gaussian component weighted $\frac{1}{3}$ in the mixture. 200 test inputs are sampled from the same Gaussians, similarly with $\frac{1}{3}$ weighting for each Gaussian. This dataset is designed to highlight a high level of heteroscedastic aleatoric uncertainty with some epistemic uncertainty at the regions in between the Gaussians.

Dataset 3 is from Tagasovska and Lopez-Paz (2019). It is generated with

$$\mathbf{y} = \sin(2\pi\mathbf{x}) + \epsilon, \quad \epsilon \sim \text{Exp}\left(\frac{1}{3}\right),$$

by uniformly sampling 800 training inputs and 200 validation inputs from $[0, 0.5] \cup [1.5, 2]$ and, after computing $\mathbf{y}$, normalizing each input $\mathbf{x}$ by subtracting the mean of the whole set $\mathbf{X}$ and dividing by the standard deviation of $\mathbf{X}$. This dataset is designed to highlight epistemic uncertainty with the gap between the two regions of training points and asymmetric aleatoric uncertainty with noise generated from an exponential distribution.

Experimental parameters

We use ReLU activation functions for all datasets. We use 1 hidden layer with 50 hidden nodes for Datasets 1 and 3 (as suggested by Yao et al. (2019) for the Dataset 1), and 2 hidden layers with 20 nodes for Dataset 2 (as suggested by Depeweg et al. (2017)).

When using Bayesian inference methods (HMC, BBB, MC dropout, NL) for both the homoscedastic and heteroscedastic noise models, we place normal priors over the weights $\mathbf{W} \sim \mathcal{N}(0, 1)$ and use the true output noise for each dataset for the homoscedastic noise model.

We run each method with 5 random restarts. Given that there are no formal metrics that formalize desiderata for aleatoric or epistemic uncertainty estimation (Yao et al., 2019), we select visuals for our results based upon which restart best captures uncertainty from an intuitive standpoint.

We discuss additional method-dependent parameters, which were selected to optimize performance, in Appendix A.

Evaluation metrics

We use RMSE and the average marginal log-likelihood as evaluation metrics for validation, as per convention in the literature. We note, however, that Yao et al. (2019) established that these metrics are not reliable indicators for quality of posterior approximation nor uncertainty approximation, and as such do not discuss them in our results, as they are not the focus of this work.

5.1.2 Results

Figures 5.1, 5.2, and 5.3 provide a comparison of all seven methods for regression on Dataset 1, 2, and 3 respectively. Note that we include the results for the regularized NL model rather than the MAP NL model in these figures, and instead include the results for the MAP NL

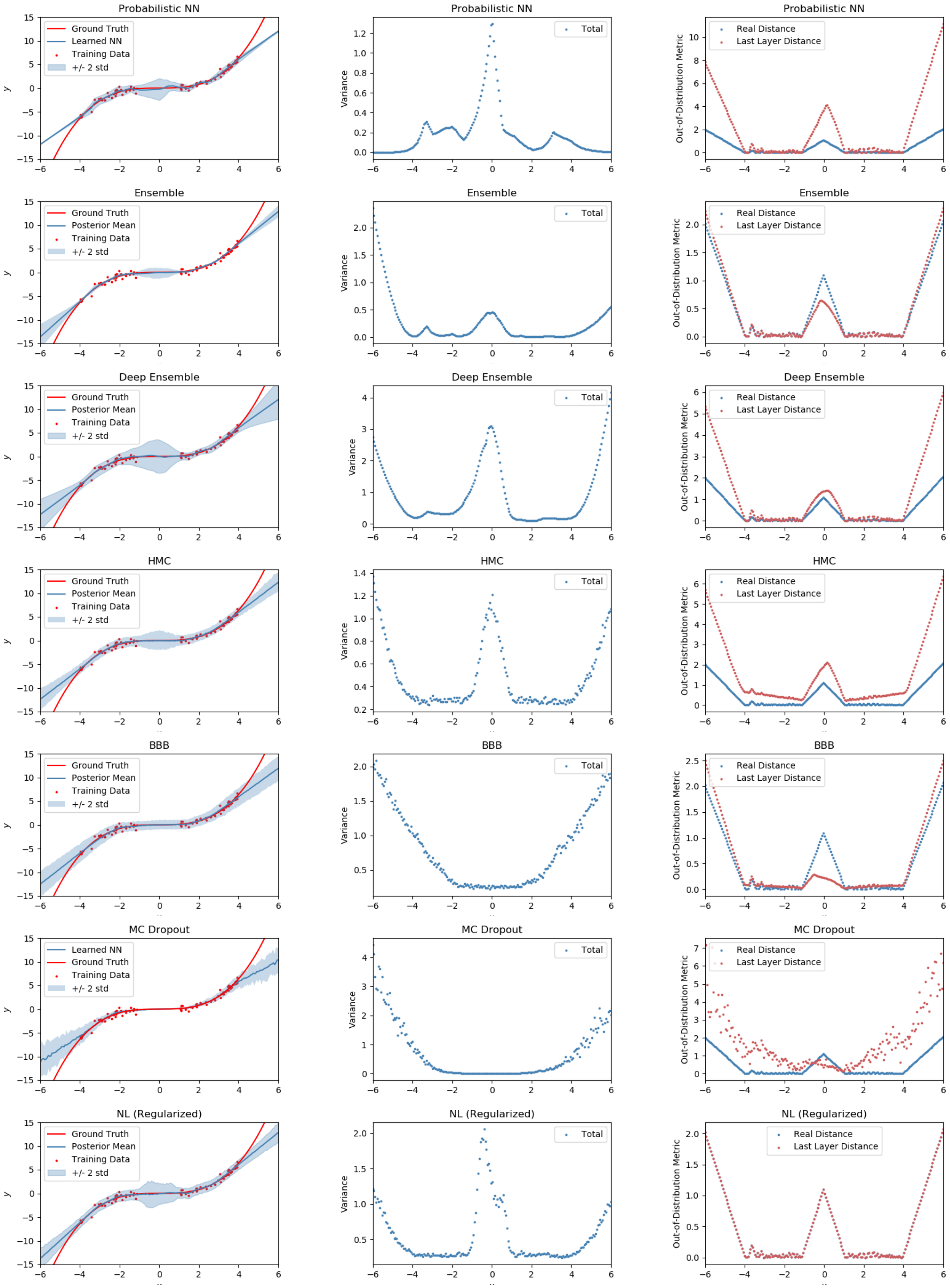


Figure 5.1 A comparison of all seven methods for regression on Dataset 1, with the probabilistic NN, ensemble, deep ensemble, and MC dropout approximating a heteroscedastic regression model and HMC, BBB, and NL approximating a homoscedastic regression model.

model in Appendix B. The first column of images displays the posterior predictive for each method, with 2 standard deviations denoted with '+/- 2 std.' The second column of images plots the variance (as represented by the '+/- 2 std' in the first column of visuals) against the x-input. The third column plots the OOD detection metrics against the x-input.

Then, Figures 5.4, 5.5, and 5.6 display the results for the LOTV decomposition of aleatoric and epistemic variance for deep ensemble and MC dropout, alongside the uncertainty estimates from the artisanal SQR and OC approaches. We discuss results method by method rather than by dataset, and then discuss the last layer distance estimates for epistemic uncertainty.

First, the probabilistic NN matches the aleatoric variance inherent in each dataset relatively well but does not capture the epistemic uncertainty well, as reflected by its narrow posterior predictive over data scarce regions for Dataset 2. This is all consistent with expectation, as the probabilistic NN only models aleatoric variance.

Second, we find that ensemble generally fails to capture any aleatoric uncertainty across all three datasets, and seems to capture some of the epistemic uncertainty, typically on the outsides of the dataset rather than the in-between region (as highlighted by Dataset 3). Moreover, we generally find that ensemble methods often produces similar solutions among the individual NNs due to initialization and optimization issues, which is consistent with the findings of Yao et al. (2019). This implies that ensemble methods are unreliable and fail to accurately capture either aleatoric or epistemic uncertainty.

By comparison, deep ensemble fares better and we find that it provides relatively accurate representations of the aleatoric and epistemic uncertainty for Datasets 2 and 3 in Figures 5.5 and 5.6 when we apply the LOTV. However, we see that it inaccurately represents the aleatoric uncertainty for Dataset 1, which we might expect to be a natural consequence given that it is comprised of probabilistic NNs designed to exclusively capture aleatoric variance. Furthermore, we find that the estimates for the epistemic are constrained within the bounds created by the aleatoric variance estimates produced by individual probabilistic NNs, which suggests that this method would be poorly equipped to capture epistemic uncertainty for datasets with low levels of homoscedastic noise (as in Dataset 1).

Next, considering HMC and examining the results for Datasets 1 and 3, we notice that the posterior predictive for HMC captures some of the epistemic uncertainty in the gap between the training points in both datasets, producing a sort of bubble in the in-between region. Given that HMC is considered in the literature a way of obtaining a ground truth approximation for the posterior and that it is applied to a homoscedastic regression model, we consider this representation desirable. By comparison, BBB does not capture any such bubble for any of the datasets, which is consistent with the literature (Yao et al., 2019). Neither HMC nor BBB reflect the heteroscedastic uncertainty implicit in Dataset 2, which

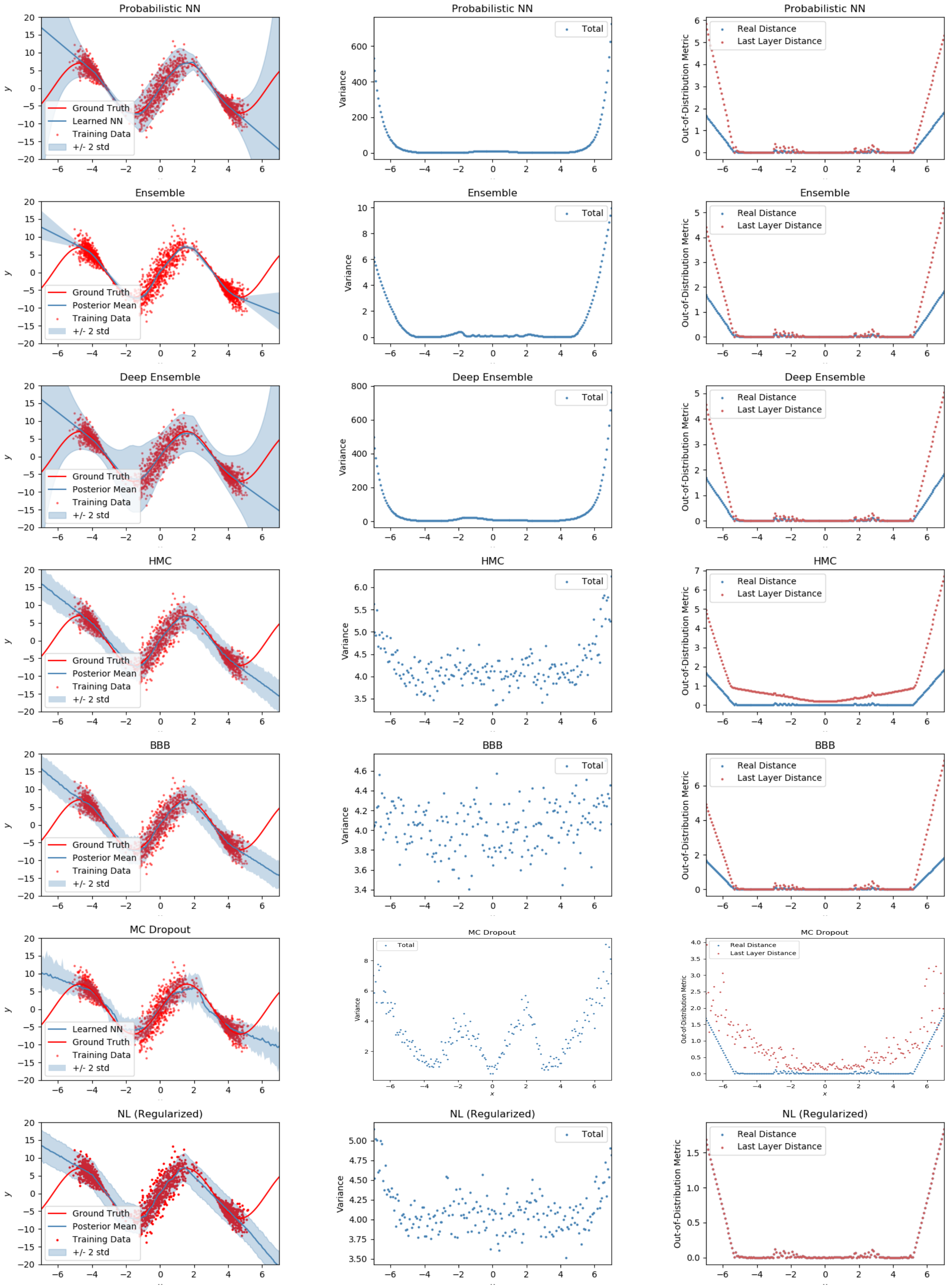


Figure 5.2 A comparison of all seven methods for regression on Dataset 2, with the probabilistic NN, ensemble, deep ensemble, and MC dropout approximating a heteroscedastic regression model and HMC, BBB, and NL approximating a homoscedastic regression model.

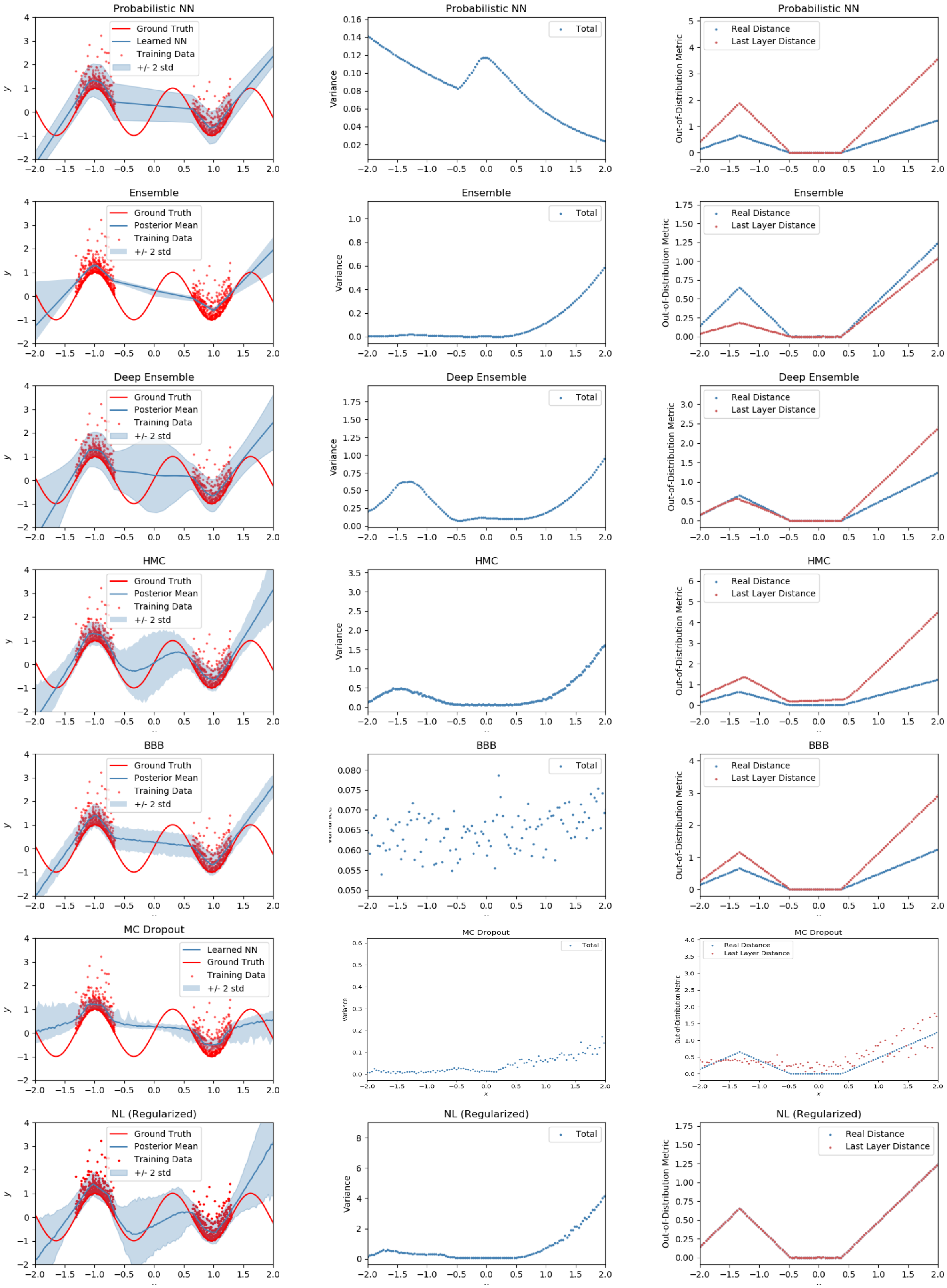


Figure 5.3 A comparison of all seven methods for regression on Dataset 3, with the probabilistic NN, ensemble, deep ensemble, and MC dropout approximating a heteroscedastic regression model and HMC, BBB, and NL approximating a homoscedastic regression model.

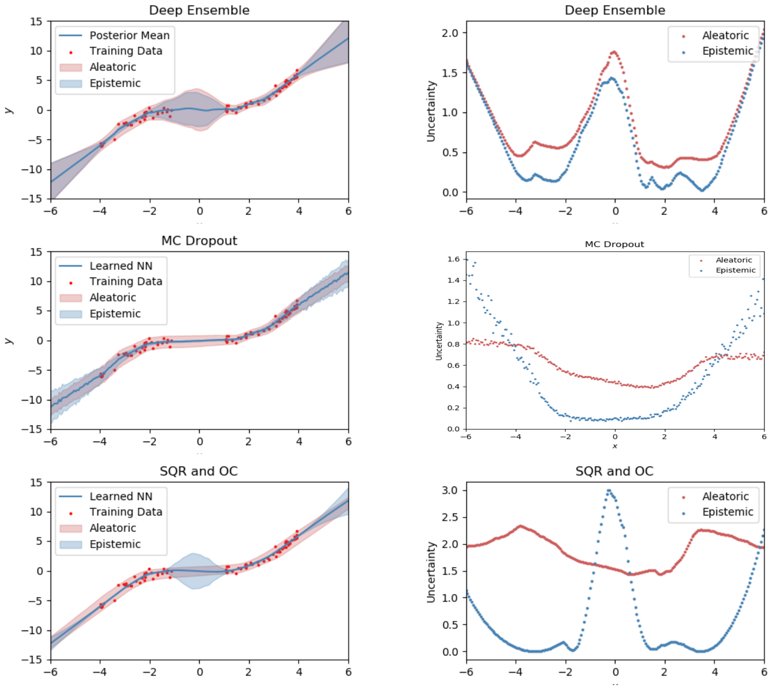


Figure 5.4 Uncertainty estimates for Dataset 1, obtained from the LOTV decomposition applied to the heteroscedastic noise model, with deep ensemble and MC dropout as inference methods, and SQR and OCs.

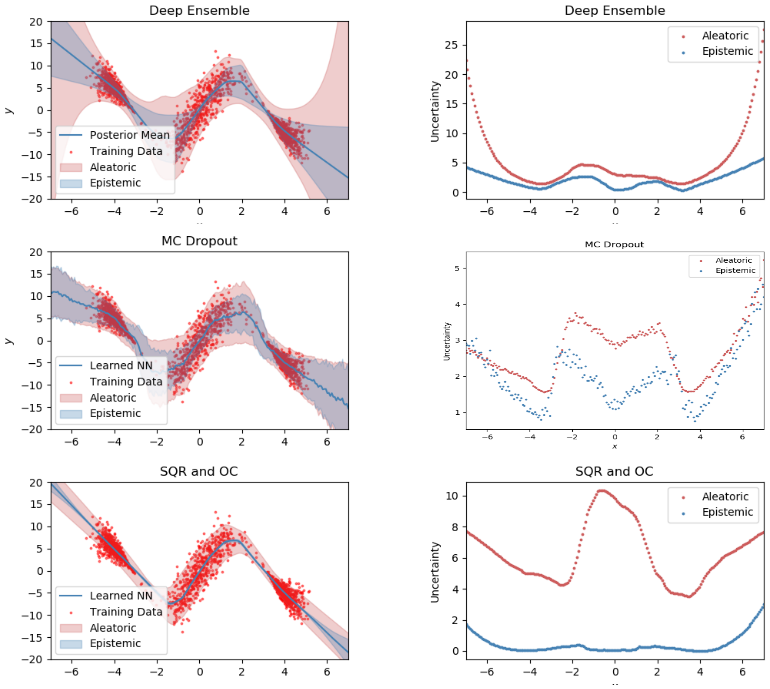


Figure 5.5 Uncertainty estimates for Dataset 2 from the LOTV decomposition applied to the heteroscedastic noise model, with deep ensemble and MC dropout as inference methods, and SQR and OCs.

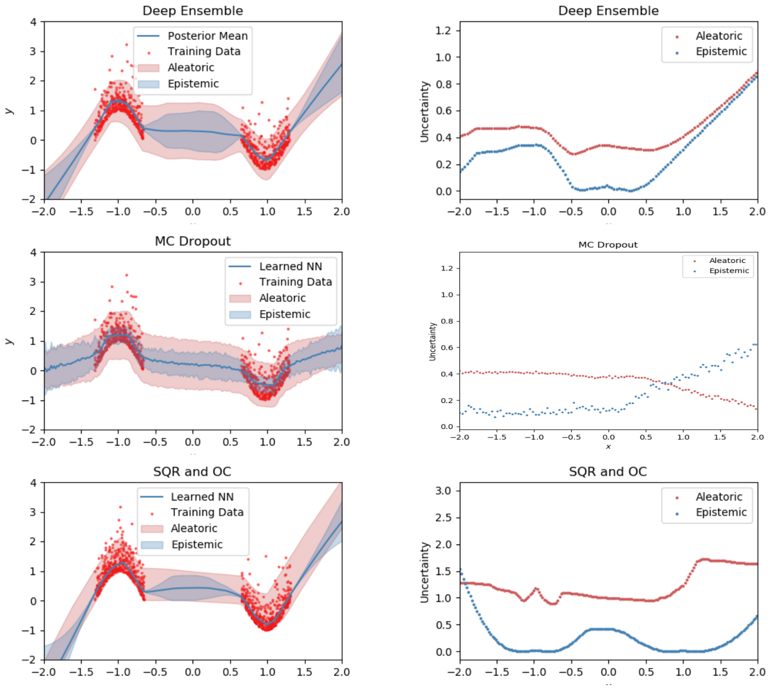


Figure 5.6 Uncertainty estimates for Dataset 3 from the LOTV decomposition applied to the heteroscedastic noise model, with deep ensemble and MC dropout as inference methods, and SQR and OCs.

is to be expected.

Examining the results for the NL model, we can see that the regularized NL model produces a posterior predictive that is rather similar to the one produced by HMC for all three datasets and similarly captures some of the in-between uncertainty for Datasets 1 and 3. We note, however, that HMC produces comparatively smoother and more symmetrical uncertainties. In this sense, we can think of the NL model as an approximation to the full Bayesian model, with assumed delta collapse of the weight posterior for the first layers on which the NL model does not place priors. This might suggest that the NL model—when constructed with a sufficiently large feature space for the last layer—might be an attractive alternative to the comparatively intractable HMC and BBB.

We note that MC dropout as displayed in Figures 5.1, 5.2, and 5.3 is applied to a homoscedastic model, and as displayed in Figures 5.4, 5.5, and 5.6 is applied to a heteroscedastic model and paired with the LOTV decomposition as proposed by Kendall and Gal (2017). However, despite performing a wide grid search over dropout probabilities p for MC dropout, we find that it—when applied to the heteroscedastic noise model—fails to represent the in-between, epistemic uncertainty in Datasets 1 and 3 well in Figures 5.4 and 5.6. While MC dropout seems to capture aleatoric uncertainty to some extent, its representation of epistemic uncertainty seems to mimic the pattern exhibited by aleatoric uncertainty in all three datasets, thereby implying that the two uncertainties are perhaps not accurately separated. Furthermore, we note that MC dropout is extremely sensitive to the dropout rate p, which is a substantial drawback when dealing with real-world datasets for which the true output noise is not known. In fact, adjusting p is in some ways comparable to directly tuning the output noise as represented by MC dropout; it is common for the method to severely overrepresent or underrepresent the uncertainty depending on the dropout rate.

We find that the last layer distance metric, which is a proxy for epistemic uncertainty, peaks (or relatively peaks, as demonstrated by the small spikes) for all methods except MC dropout in regions of data scarcity for Datasets 1 and 2. However, last layer distance does not accurately reflect epistemic uncertainty for Dataset 3, where it peaks somewhere near the training points rather than in the in-between region, perhaps due to the asymmetric nature of the exponentially generated noise.

Finally, we see that the SQR and OCs implementation accurately captures the aleatoric and epistemic uncertainty in Datasets 1 and 3. However, OCs do not do as well on Dataset 2, where they estimate close to zero epistemic uncertainty for the regions of relative data scarcity between the means of the Gaussians. This is consistent with expectation, as OCs are designed to map epistemic uncertainty to zero near the training distribution, but nevertheless reflects a shortcoming of the approach—that it cannot distinguish between relative levels of epistemic uncertainty, as demonstrated by its equal epistemic uncertainty estimate of zero for a region of scarce data and a region of extremely high data density in Dataset 2.

Given that SQR and OCs are artisanal approaches—and not model-agnostic or method-agnostic ones as we seek—we should be wary of directly comparing their representations of uncertainty compared to the previously discussed methods.

5.2 Classification

5.2.1 Experimental setup

Given a set of N observations of inputs and outputs $\{(\mathbf{x}_1, \mathrm{y}_1),..., (\mathbf{x}_N, \mathrm{y}_N)\}$ where $\mathbf{x}_i \in \mathbb{R}^2$ and the class label $\mathbf{y}_i \in \{0, 1\}$, we consider a model where $\mathbf{y} = \sigma\left(\mathbf{f}^{\mathbf{W}}(\mathbf{x})\right)$, where $\mathbf{f}^{\mathbf{W}}$ is our NN parameterized by $\mathbf{W}$ and σ is the softmax function as defined in Equation 2.7 (or the sigmoid, as softmax is a generalization of the sigmoid, which is used for binary classification).

In addition to this conventional model that is very commonly used in the literature, we also consider the heteroscedastic classification model proposed by Kendall and Gal (2017) (§3.2.3), where our NN is split to predict $\mathbf{f}^{\mathbf{W}}(\mathbf{x})$ and the intermediate heteroscedastic variance $\mathbf{g}^{\mathbf{W}}(\mathbf{x})$.

Methods

We pair five methods with the conventional model described above: a deterministic NN, an ensemble of deterministic NNs, HMC, BBB, and MC dropout. We use one method, MC dropout, with the heteroscedastic classification model, as proposed by Kendall and Gal (2017), given that HMC and BBB are not equipped to tractably approximate its posterior.

Uncertainty metrics

We consider several different kinds of uncertainty estimates.

First, for each of the methods, we visualize the variance of the posterior predictive (two predictive standard deviations), which is not technically an uncertainty estimate but arises from the model assumptions and heavily informs the computation of all the other metrics. Since this represents the variance that arises from model sampling (§3.2.2), this might technically be considered an estimate for epistemic uncertainty but more realistically is reflective of total predictive uncertainty for reasons similar to those discussed above for regression.

Second, we apply all of the statistical measures of variability discussed in §3.3.1 (variation ratio, predictive entropy, Gini-Simpson index, and Allaj index) to obtain total uncertainty estimates. We then decompose the latter three estimates into their aleatoric and epistemic components using the decomposition that is based upon the statistical principle of conditioning on $\mathbf{W}$, as discussed in §3.3.2.

Third, we compute estimates for aleatoric and epistemic uncertainty based upon the three LOTV decompositions proposed by Gal (2016), Kwon et al. (2018), and Shridhar, Laumann, and Liwicki (2018) (§3.3.2). We note that the first decomposition by Gal (2016), as discussed in §3.2.3, is only paired with the heteroscedastic classification model (and, by extension, only with MC dropout), whereas the other two are applied to the conventional classification model.

Fourth, we compute four OOD detection metrics discussed in §3.3.3—last layer distance, largest softmax score, functional margin, and Gaussian covariance—as estimates for epistemic uncertainty.

Datasets

We use three synthetic datasets. For all three datasets, we sample 80 training inputs, 20 validation inputs, and 100 test inputs equally from two class targets $c \in \{0, 1\}$.

Dataset 1 is from Yao et al. (2019). We sample two balanced class targets from two multivariate Gaussian distributions,

$$p_1 \sim \mathcal{N}\left(\begin{bmatrix}2\\2\end{bmatrix}, \mathbf{I}\right),\ p_2 \sim \mathcal{N}\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \mathbf{I}\right).$$

In this dataset, the classes used to construct the training data are clearly separated and a clear decision boundary—or many—can be drawn between the two classes. We might consider the region between the two classes, which lacks data, as having high epistemic uncertainty, but it is also true that the regions that are outside of the classes in other directions (i.e., not between the two classes) also represent domains with high epistemic uncertainty due to the lack of data.

For Dataset 2, we sample two balanced class targets from two multivariate Gaussian distributions that are more closely placed together,

$$p_1 \sim \mathcal{N}\left(\begin{bmatrix}1\\1\end{bmatrix}, \mathbf{I}\right),\ \ p_2 \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \mathbf{I}\right).$$

This dataset is designed to exhibit considerable class overlap, which—in the context of classification—can be interpreted as aleatoric uncertainty, as it is irreducible uncertainty that exists as an inherent property of the data.

For Dataset 3, we sample two balanced class targets from four multivariate Gaussian distributions,

$$\begin{gathered} p_1 \sim \mathcal{N}\left(\begin{bmatrix}2\\2\end{bmatrix}, \mathbf{I}\right),\quad p_2 \sim \mathcal{N}\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \mathbf{I}\right), \\ p_3 \sim \mathcal{N}\left(\begin{bmatrix}2\\-2\end{bmatrix}, \mathbf{I}\right),\ p_4 \sim \mathcal{N}\left(\begin{bmatrix}-2\\2\end{bmatrix}, \mathbf{I}\right), \end{gathered}$$

where p_1 and p_2 correspond to the first class and p_3 and p_4 correspond to the second class. This dataset is constructed such that the two classes have multiple clusters each and therefore cannot be separated with a single linear decision boundary. Epistemic uncertainty peaks outside of the four Gaussian clusters of training data and aleatoric uncertainty peaks near the class boundaries in between the clusters.

Experimental parameters

We use ReLU activation functions for all classification tasks but, naturally, apply a softmax function to the last layer output to produce a probability estimate. We use 2 hidden layers with 10 hidden nodes each (as suggested by Yao et al. (2019) for Dataset 1) for all classification tasks. Otherwise, our experimental parameters are identical to those for synthetic regression, and tuning for all other method-dependent parameters is discussed in Appendix A.

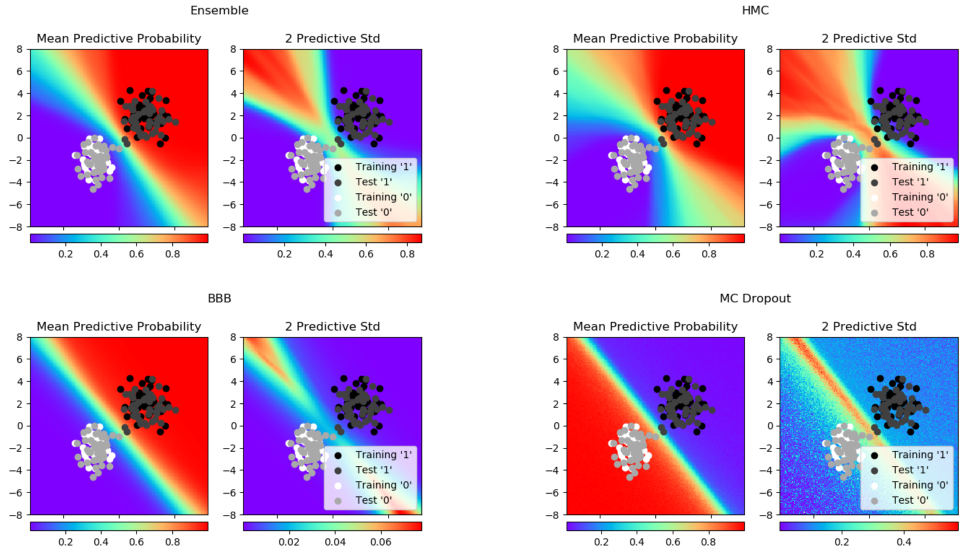


Figure 5.7 The mean predictive probability and two standard deviations of the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 2.

Evaluation metrics

We use the average marginal log-likelihood as an evaluation metric for validation, but—similar to synthetic regression—do not place emphasis on this metric and instead focus evaluation on the uncertainty metrics described earlier.

5.2.2 Results

We visualize the mean predictive probability and two standard deviations of the posterior predictive for the ensemble, HMC, BBB, and MC dropout for all three datasets in Figures 5.7, 5.8, and 5.9 respectively. We visualize the mean predictive probability for the deterministic NN—to which our decompositions of aleatoric and epistemic uncertainty cannot be applied—in Appendix B.

Based on these figures, it appears that ensemble and HMC seem to produce the best posterior predictives for each dataset. In particular, both of these methods express wider possible ranges for the linearized decision boundary between the two classes, as seen per the more gradual mean predictive probability and the relatively large predictive standard deviations. We can interpret this as indicating that these methods are better equipped to model epistemic uncertainty, given that they consider a range of possible thresholds in regions with no data. By comparison, BBB and MC dropout express relatively thin ranges for the decision boundaries with low predictive standard deviations—and therefore fail to

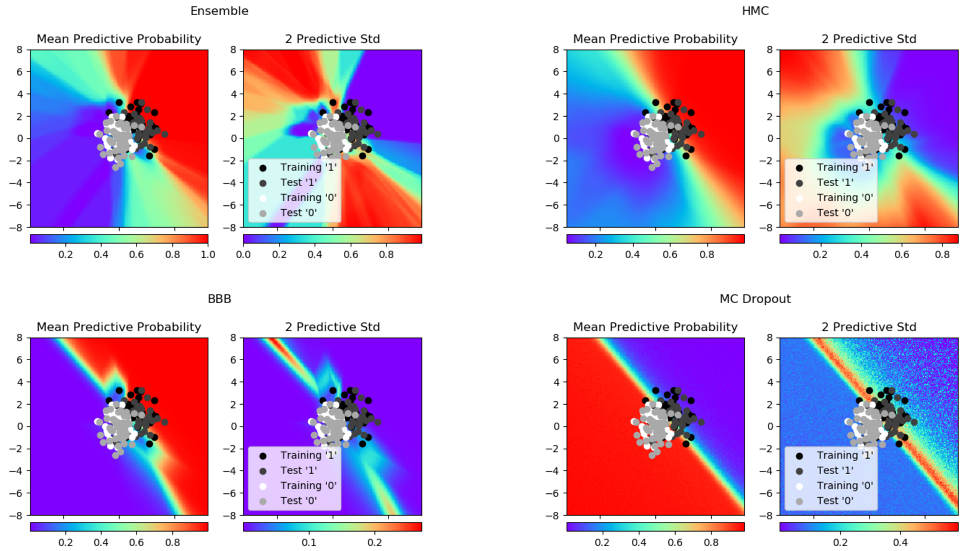


Figure 5.8 The mean predictive probability and two standard deviations of the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 2.

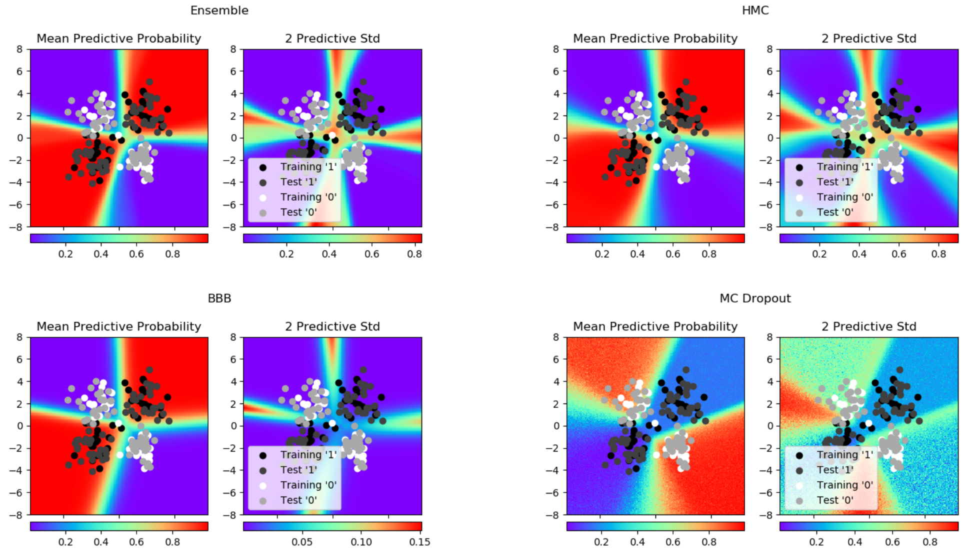


Figure 5.9 The maen predictive probability and two standard deviations of the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 3.

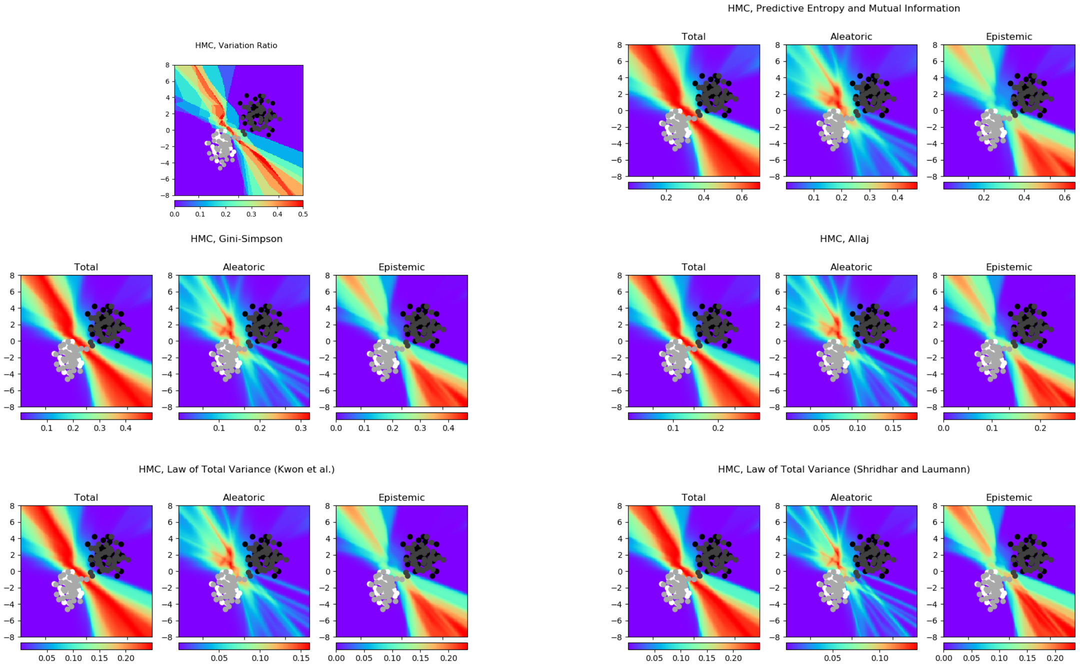


Figure 5.10 Statistical measures of variability applied to the posterior predictive of the conventional classification model, with HMC as the inference method, for Dataset 1, with measures decomposed into aleatoric and epistemic components by conditioning on $\mathbf{W}$.

capture epistemic uncertainty. This is the case for all three datasets, and we see that MC dropout most consistently produces a thin, linear decision threshold, suggesting that it is limited in its expressiveness.

We highlight the different statistical measures for variability in Figure 5.10, in which we apply them to the posterior predictive produced by HMC for Dataset 1. It is evident from this figure that all statistical variability metrics produce functionally similar estimates for aleatoric and epistemic uncertainty. Generally speaking, aleatoric uncertainty is considered to correspond to regions of the data domain that exhibit class overlap, and epistemic uncertainty is considered to correspond to regions over which the decision boundary is variable. We note, however, that none of these metrics predict high levels of epistemic uncertainty for domains well within the decision threshold, with the single exception of predictive entropy and mutual information when it is applied to the posterior predictive produced by HMC for Dataset 2 (Figure 5.11).

Given the similarity between all statistical measures of variability, we limit the metrics that we visualize in the remainder of this section to predictive entropy and mutual informa-

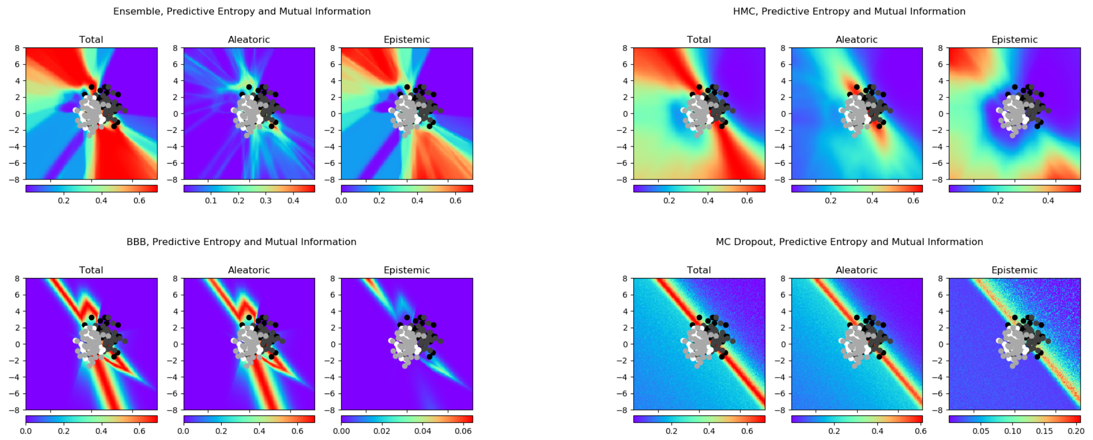


Figure 5.11 Predictive entropy and mutual information applied to the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 2.

tion out of convention.[1]. Indeed, because it seems that selection of inference method—as opposed to metric—primarily informs the computation of uncertainty estimates, we focus on visual comparisons of estimates for predictive entropy and mutual information produced by the posterior predictives for different methods.

We visualize predictive entropy and mutual information for Datasets 2 and 3 in Figures 5.11 and 5.12 respectively. In each of these subfigures, the leftmost column is the predictive entropy and represents total uncertainty, the middle column is the aleatoric component as computed in Equation 3.17, and the rightmost column is mutual information as computed in Equation 3.18 and represents epistemic uncertainty.

Next, we compute our four OOD metrics for the conventional classification model, as approximated with each of the four inference methods, for all three datasets. We visualize these metrics for Datasets 2 and 3 in Figures 5.13 and 5.14 respectively and for Dataset 1 in Appendix B.

Examining Figures 5.13 and 5.14, we notice that last layer distance and Gaussian covariance capture epistemic uncertainty in a fashion that none of our statistical decompositions do. In other words, they produce high estimates for epistemic uncertainty in regions that are far removed from the data, producing a circular ring around the data as opposed to high uncertainty estimates at class boundaries. Although such estimates may be less helpful for simple, pedagogical datasets that have clearly separable classes, they are potentially quite useful for classification tasks that involve complex, high dimensional inputs with classes that are not immediately separable. Image classification, which we evaluate in §6.2.3 is one such

[1] Note that we include visualizations of the LOTV as proposed by Kwon et al. (2018) in Appendix B

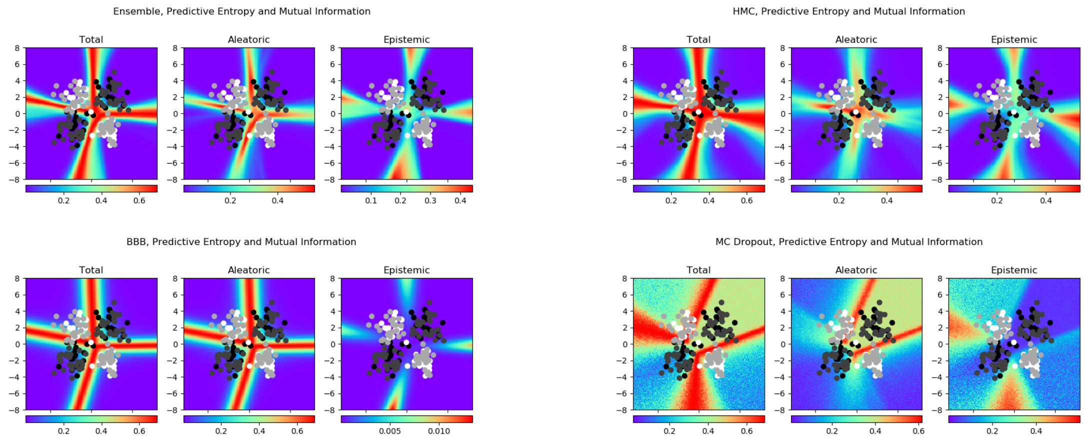


Figure 5.12 Predictive entropy and mutual information applied to the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 3.

example of a task.

Furthermore, we note that largest softmax score and functional margin peak in uncertainty for lower values; for these two metrics, higher values reflect greater certainty (as previously discussed in §3.3.3). We also notice that they have similar properties to the statistical decompositions that we have considered, likely because they are also immediate functions of the model logit output.

We have hitherto only considered the conventional classification model. For comparison, we evaluate the heteroscedastic classification model proposed by Kendall and Gal (2017) on all three datasets in Figure 5.15, using MC dropout as an inference method and the LOTV decomposition from §3.2.3 to estimate aleatoric and epistemic uncertainty. In general, we find that the model produces narrow, linear decision boundaries for the first two datasets, similar to the posterior predictive provided by MC dropout for the conventional classification model.

Moreover, the uncertainty estimates produced by the model are unlike anything seen up to this point. The model predicts a large, broad band of aleatoric uncertainty and a thin, narrow band of epistemic uncertainty—both of which are located at the decision boundary thresholds—for Datasets 1 and 2. The broad aleatoric uncertainty band can be attributed to the heteroscedastic regression noise that is artifically injected at an intermediate level in the model, which yields a high level of aleatoric variance that is not inherently found in the data itself. Given this, it is unclear from an interpretability standpoint as to how we should treat the estimates for aleatoric uncertainty for this model class.

Finally, we compute AUC as a function of the proportion of retained data in Figure

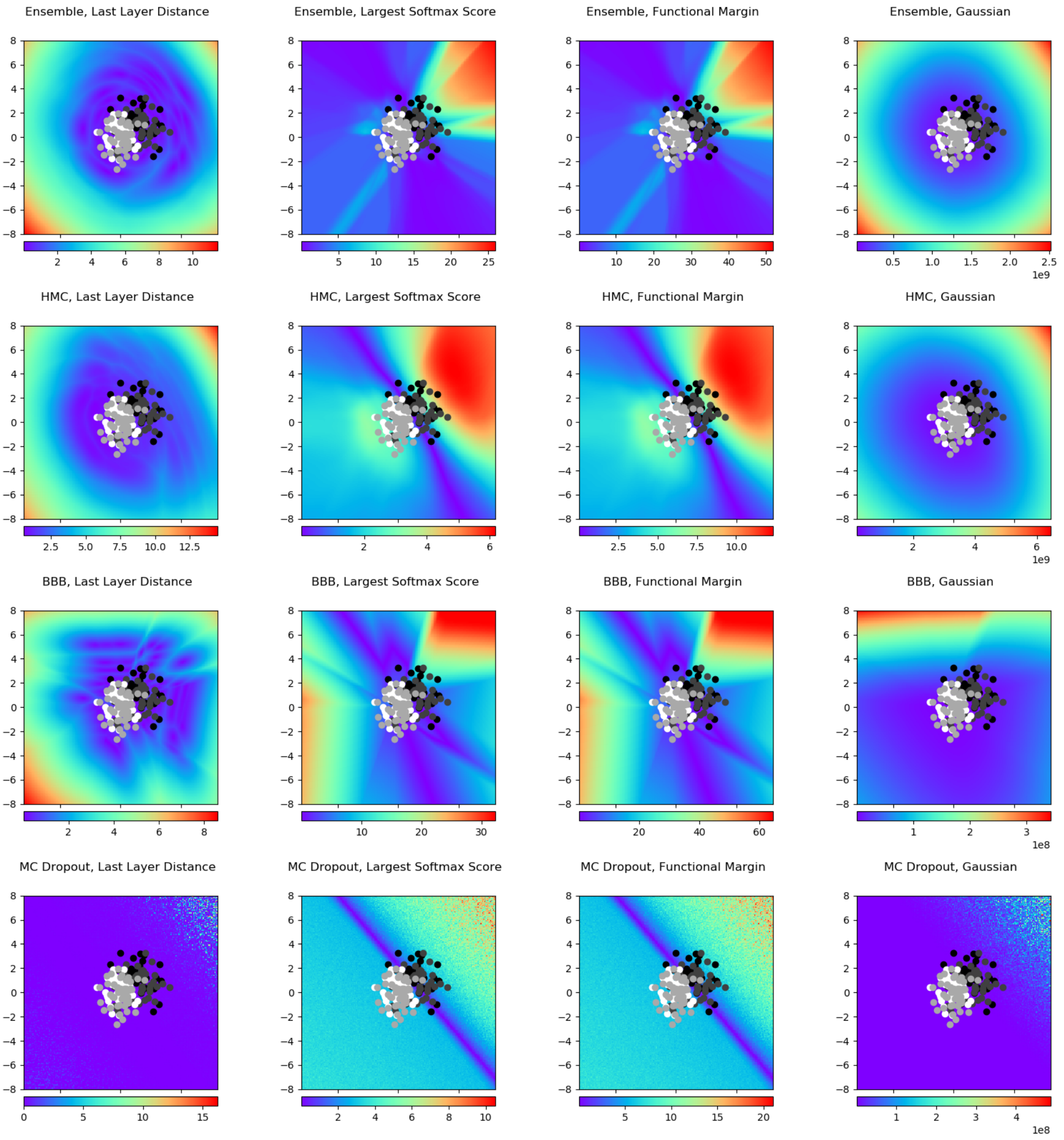
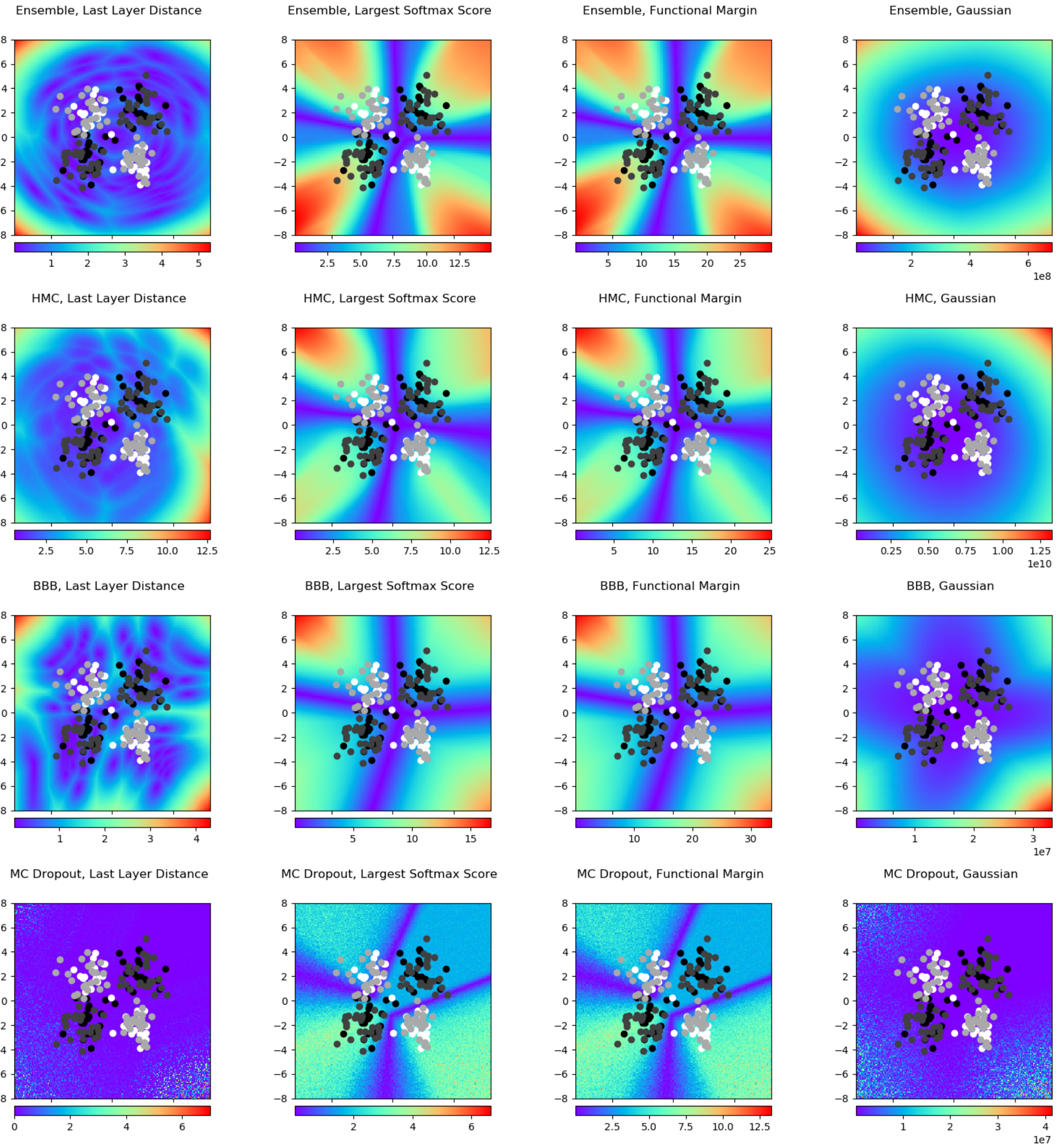


Figure 5.13 OOD metrics applied to the last layer (last layer distance, Gaussian) and output layer (largest softmax score, functional margin) representations of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 2.


Ensemble, Last Layer Distance

Ensemble, Largest Softmax Score

Ensemble, Functional Margin

Ensemble, Gaussian

HMC, Last Layer Distance

HMC, Largest Softmax Score

HMC, Functional Margin

HMC, Gaussian

BBB, Last Layer Distance

BBB, Largest Softmax Score

BBB, Functional Margin

BBB, Gaussian

MC Dropout, Last Layer Distance

MC Dropout, Largest Softmax Score

MC Dropout, Functional Margin

MC Dropout, Gaussian

Figure 5.14 OOD metrics applied to the last layer (last layer distance, Gaussian) and output layer (largest softmax score, functional margin) representations of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 3.

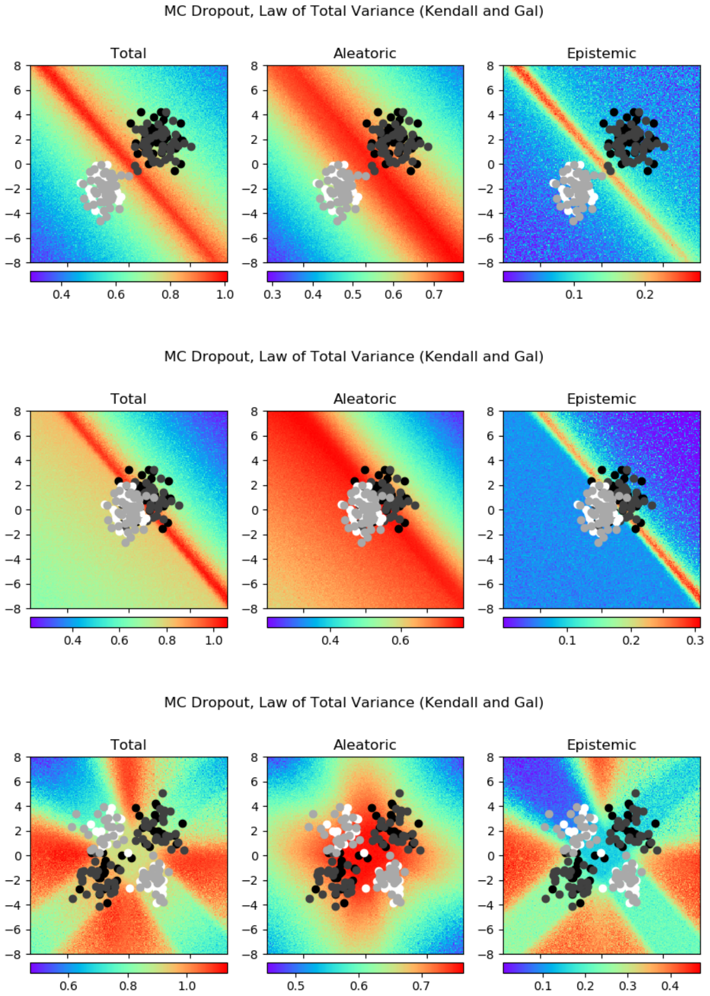


Figure 5.15 LOTV applied to the posterior predictive of the heteroscedastic classification model, with MC dropout as an inference method, as proposed by Kendall and Gal (2017), for Datasets 1, 2, and 3.

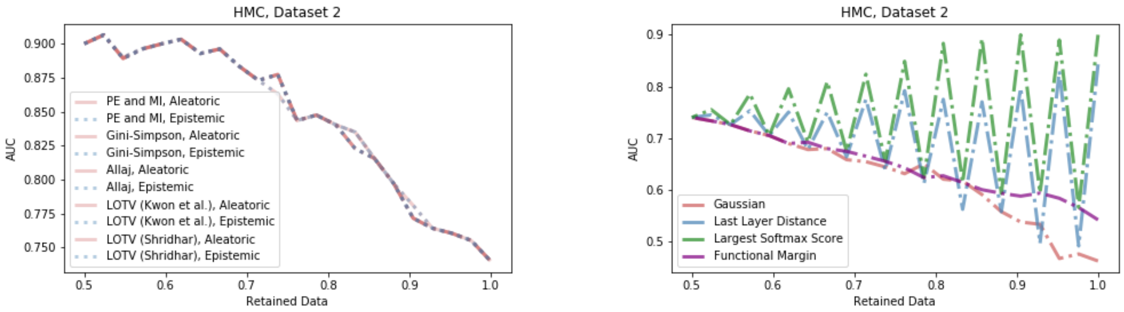


Figure 5.16 AUC vs. retained data based upon different uncertainty decompositions and OOD metrics applied to the conventional classification model, with HMC as the inference method, for Dataset 2.

5.16 for Dataset 2, which exhibits the greatest level of class overlap of the three synthetic datasets. In general, the premise behind this plot is that the methods that best capture uncertainty score better when less data is retained, deferring the least certain cases.

Based upon the left subplot in Figure 5.16, we find that all the uncertainty decompositions—including their subcomponents—produce exactly equivalent curves, achieving a maximum AUC of 0.9 when half of the data is retained. This validates our earlier findings from Figure 5.10 that all the statistical measure of variability are functionally similar and thus that predictive entropy is sufficient as one such estimate.

In the right subplot in Figure 5.16, we notice a steady increase in the AUC curve for Gaussian and functional margin, and a sharp, jagged climb in the AUC for largest softmax score and last layer distance. We theorize that this zig-zag pattern exists because data points corresponding to opposing classes are included as the proportion of retained data is increased, causing the AUC to alternate between high and low values as members of the favorable and unfavorable class are included. Although the AUC curve is less desirable for the OOD detection metrics than for the measures of statistical variability, we also recognize that this is a consequence of the design of our synthetic datasets, which included strongly clustered groups of points generated by Gaussians. Rather, in order to properly test the performance of OOD detection metrics, it is necessary to use a more complex real-life classification task that involve epistemic uncertainty, such as the MNIST task we use in §6.2.3.

Chapter Six

Evaluation on Real Data

Evaluating different approaches for quantifying uncertainty on synthetic data is instructive because it allows us to compare estimates relative to the true parameters or properties of a particular dataset, which in turn exposes flaws in the models, methods, and metrics that we pair together to arrive at estimates for uncertainty. By comparison, modeling uncertainty on real data is considerably more challenging due to our ignorance about the true parameters that underlie the data-generating process in the real world. Thus, while we acknowledge that application to real data is a critical and warranted component for evaluation of both methods and metrics in machine learning literature, we also note that under the scope of this thesis it is necessary to tread carefully when interpreting real-world experimental results in an evaluative fashion. This is primarily so not only because of our lack of knowledge (epistemic uncertainty, one might say) of the processes underlying different regions in the data but also because we lack metrics that formalize that which is desirable for aleatoric and epistemic uncertainty estimates in the first place (as compared to other real-world tasks that have clearly delineated objectives and standards).

As such, while this chapter is nevertheless an important part of this thesis, it is important to frame it as complementary (and perhaps secondary) to §5, in that making evaluative statements about model and metric performance is difficult—unfounded perhaps—in the absence of existing norms for the aleatoric and epistemic uncertainty when working with real data.

This being said, however, we approach the experiments within this chapter from a practical standpoint, seeking first to understand whether our previously discussed metrics can capture notions of human uncertainty and second to evaluate which metrics can improve performance when paired with an abstention criteria, as discussed in AngelosFilos, Gomez, and Rudner (n.d.) in the context of a diabetic retinopathy diagnosis task.

In this chapter, we use two standard machine learning benchmarks: the Boston Housing dataset from the UCI Machine Learning Repository (Dua and Graff, 2017), on which we perform regression, and the MNIST handwritten digit dataset (LeCun, Cortes, and Burges, 1998), on which we perform classification.

6.1 UCI regression

6.1.1 Experimental setup

Because real-life datasets, including the Boston Housing dataset, typically exhibit considerable variation in the level of noise across different parts of the data domain, we use the more expressive heteroscedastic regression model, where (as previously discussed in §5.1.1) we assume a likelihood of the form $\mathbf{y}_i = \mathcal{N}(\mathbf{f}^{\mathbf{W}}(\mathbf{x}_i), \sigma^2(\mathbf{x}_i))$ and model aleatoric noise (also known as model precision) as the data-dependent term $\sigma^2(\mathbf{x}_i)$. We select MC dropout as our inference method given that it is the only tractable Bayesian approximation for this model and that non-Bayesian alternatives—aside from deep ensembles—do not offer a way to explicitly model nor attain estimates for aleatoric and epistemic uncertainty (which is the focus of this work).

Although we would ideally also implement and evaluate deep ensembles in addition to MC dropout for this task, we reserve this for future work and instead practically consider this experiment as an evaluation of Kendall and Gal (2017)'s combined heteroscedastic aleatoric and epistemic uncertainty model (§3.2.3). Given this, we also apply MC dropout to a homoscedastic variance model for comparative purposes to see the empirical difference between Kendall and Gal (2017)'s proposed model and the more general MC dropout approach.

Datasets

The Boston Housing dataset contains information collected by the U.S. Census regarding housing near Boston, Massachusetts and contains 506 samples and 13 features, which are used to predict the median value of a home. Instead of using the standard training split for the Boston Housing dataset that is proposed in the literature (Hernández-Lobato and Adams, 2015; Bui et al., 2016; Mukhoti, Stenetorp, and Gal, 2018), which is not informative for evaluating estimates of aleatoric and epistemic uncertainty, we instead perform regression on a variant of this dataset that is designed to test for in-between uncertainty.

We adapt the protocol proposed in Foong et al. (2019) to create such a split. For each of the 13 input dimensions of $\mathbf{x}_n \in \mathbf{R}^{13}$, we sort the datapoints in increasing order as per that dimension and then remove the middle $\frac{1}{3}$ of the datapoints from the training data. However, whereas Foong et al. (2019) uses the outside $\frac{2}{3}$ of the points as the training data and the middle $\frac{1}{3}$ as the test data, we instead take 20% of the data points from each domain (left, middle, and right) for testing and 80% of the data points from the left and right domains

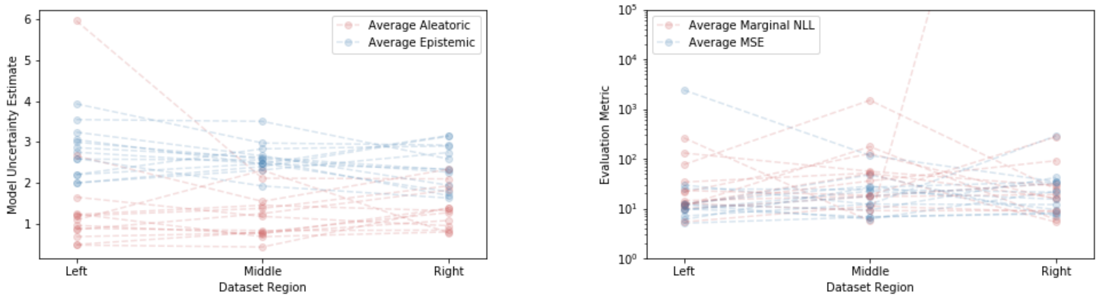


Figure 6.1 Average aleatoric and epistemic uncertainty estimates and evaluation metrics produced by the heteroscedastic uncertainty model, with MC dropout as an inference method, for the different regions of the 13 Boston Housing gap datasets.

for training. This allows us to compute both evaluation and uncertainty metrics for each of the domains (left, middle, and right) to assess whether the model captures a higher level of epistemic uncertainty in the middle, gap domain.

Experimental parameters

We use a similar NN architecture to the one constructed for Datasets 1 and 3 in synthetic regression. We use exclusively ReLU activation functions and use 1 hidden layer with 50 nodes, but adapt the input layer to handle 13 input features. We use normal priors over the weights $\mathbf{W} \sim \mathcal{N}(0, 1)$ and use output variance of 9 for the homoscedastic model.

Evaluation metrics

We compute the MSE, average marginal log-likelihood, and average aleatoric and epistemic uncertainty on the entire test set for the full dataset regression task and for each of the three test sets corresponding to the three data regions (left, middle, and right) for the gap dataset regression task.

6.1.2 Results

We plot the evaluation metrics for the heteroscedastic uncertainty model and, for comparison, the homoscedastic uncertainty model, both paired with MC dropout, in Figures 6.1 and 6.2 respectively. Note that each of the dotted lines (both blue and red) corresponds to one of thirteen gaps created in the Boston Housing dataset.

Examining Figure 6.1, we notice that the heteroscedastic model, on average, does not produce higher epistemic uncertainty estimates for the middle region with in-between uncertainty for the gap datasets. Rather, its epistemic uncertainty estimates are about comparable

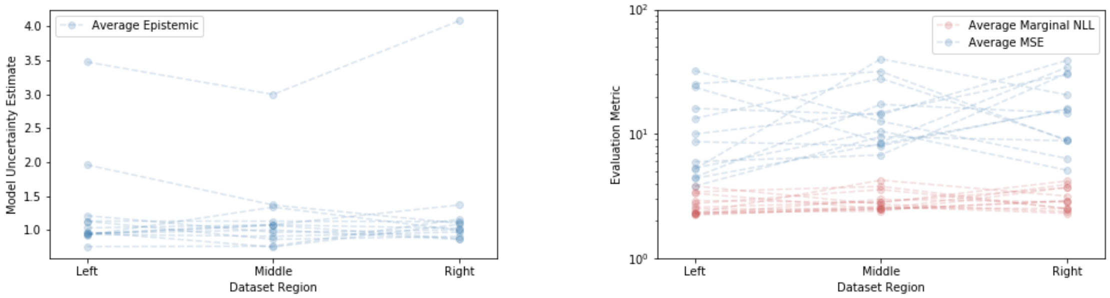


Figure 6.2 Average aleatoric and epistemic uncertainty estimates and evaluation metrics produced by the homoscedastic uncertainty model, with MC dropout as an inference method, for the different regions of the 13 Boston Housing gap datasets.

for each of the data regions, and aleatoric uncertainty increases slightly as we move from left to right. Evaluation metrics also remain roughly constant across the three regions.

We can interpret this in a few ways. First, it is likely that creating a gap corresponding to one input dimension is insufficient to create substantial epistemic uncertainty. Given that the gap is created based only upon one input dimension, twelve other data dimensions remain, likely with a density of training points comparable to prior the creation of the in-between gap (as they are not necessarily sorted in order). This likely provide sufficient information to recover the certainty lost from constructing a gap based on one dimension. As such, the design of this experiment is likely inadequate for testing for epistemic uncertainty, especially when compared to the immediacy of the gaps created in the synthetic regression datasets. This suggests the need for a better benchmark for testing for in-between uncertainty.

Furthermore, based upon the results of the synthetic regression experiments, in which MC dropout struggled to express greater epistemic uncertainty—or greater uncertainty at all—for the in-between regions, it is therefore unlikely that it is able to capture the epistemic uncertainty in an expressive way for a real-life application.

6.2 MNIST classification

We consider a conventional classification model, as well as the heteroscedastic classification model proposed by Kendall and Gal (2017), for binary classification on MNIST, leaving evaluation of multiclass classification for future work.

6.2.1 Experimental setup

With the exception of the choice of dataset, experimental parameters, and evaluation metrics, our experimental setup—including our selection of models, methods, and uncertainty metrics—is identical to that of the binary classification tasks for the synthetic datasets, as described in §5.2.1.

Dataset

The full MNIST handwritten digit dataset contains 60,000 training images and 10,000 testing images, each of which is 28×28 pixels. In order to allow for evaluation of HMC and BBB—computationally costly inference methods—we downsample the images to 5×5 pixels to reduce the input dimensionality $\sim 31\times$. We furthermore only select two images at a time for binary classification and train on only 5% of those digits, in part to decrease computational load and improve tractability and in part to allow for our models to express epistemic uncertainty for previously unseeen inputs.

We consider two binary classification tasks. The first is classification of digit class 1 vs. 2, which we will refer to as Dataset 1; this task is supposed to be indicative of classes that have relatively distinct features and should, in theory, have a clearer decision boundary. The second is classification of digit class 0 vs. 9, which we will refer to as Dataset 2; this task is representative of classes that are more similarly represented and, by extension, should be less easily separable than Dataset 1.

Experimental parameters

We use ReLU nonlinearities (except for the output layer) and 1 hidden layer with 50 nodes for both the conventional and heteroscedastic classification models. Parameters are otherwise identical to those for synthetic classification, and additional method-dependent tuning is discussed in Appendix A.

6.2.2 Evaluation metrics

We compute the average marginal log-likelihood, binary classification accuracy, and AUC. We primarily consider binary classification and AUC as a function of data when retained based upon different uncertainty estimates. Note that we focus on AUC and, as such, do not visualize binary classification in the results below, as it does not provide any additional insights.

6.2.3 Results

We compute and visualize AUC as a function of retained data for the 0 vs. 9 digit classification task in Figure 6.3, where we consider all of our different decompositions and OOD

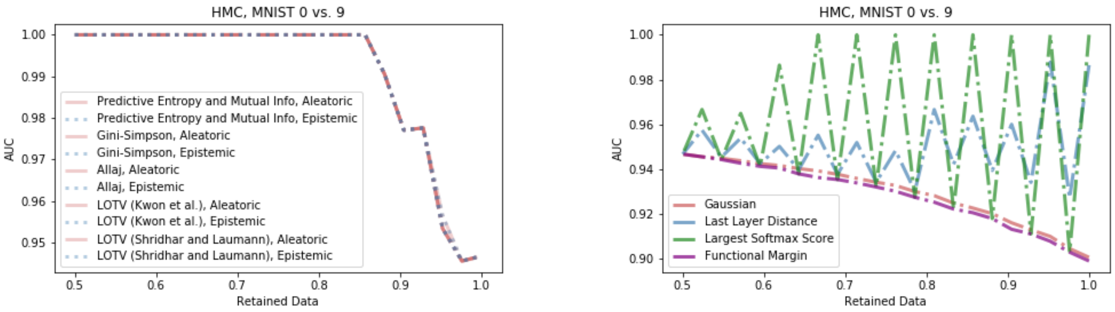


Figure 6.3 AUC vs. retained data based upon different uncertainty decompositions and OOD metrics applied to the conventional classification model, with HMC as the inference method, for MNIST Dataset 2.

detection metrics as computed by HMC. Examining this figure, we identify similar findings to those from synthetic classification, where we find that all of the decompositions for aleatoric, epistemic, and total uncertainty based upon statistical measures of variability are functionally equivalent. Because this implies that it is unnecessary to consider the estimates produced by other metrics and their decompositions, we instead we focus our attention to the effect of pairing different inference methods with predictive entropy and the OOD detection metrics.

In Figure 6.4, we compare the AUC curve that arises from estimates for predictive entropy (PE) and Gaussian covariance for each of the methods, as applied to both classification tasks. We find that HMC and MC dropout paired with predictive entropy tie for the best performance, closely followed thereafter by BBB with predictive entropy, although the slope, oddly enough, is slightly skewed. Ensemble, by comparison, severely underperforms, perhaps suggesting an instance of poor initialization or optimization.

Furthermore, we also find that the Gaussian covariance OOD metric does not necessarily produce a desirable AUC vs. retained data curve for the 1 vs. 2 digit classification task, but does so for the 0 vs. 9 retained data curve. This makes sense intuitively because the 1 and 2 digit classes likely have larger differences between their latent representations than the 0 and 9 classes, which the Gaussian covariance picks up on. Therefore, we intuit that Gaussian covariance can more readily detect input images that depart from the representation for the 0 and 9 digit classes as compared to the differing representations for the 1 and 2 digit classes.

Moreover, in the right subimage in this figure, we plot the LOTV from the heteroscedastic classification model with MC dropout, and surprisingly find that it achieves the best performance of all metrics considered. We include additional plots of AUC vs. retained data for metrics not visualized in this section in Appendix B.

Finally, in Figure 6.5, we visualize the relationship between two uncertainty metrics—

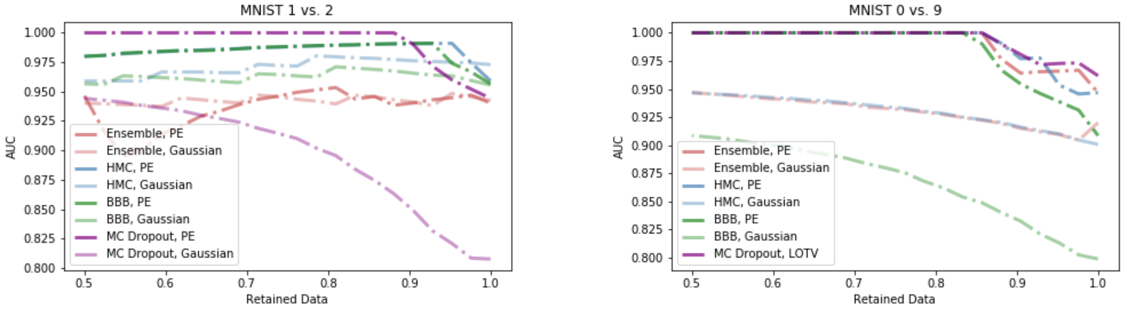


Figure 6.4 AUC vs. retained data based upon predictive entropy (PE) and the Gaussian covariance OOD detection metric applied to the conventional classification model, with ensemble, HMC, BBB, and MC dropout as the inference methods, for MNIST Datasets 1 and 2. The LOTV for the heteroscedastic classification model with MC dropout is also included in the figure on the right.

predictive entropy and the Gaussian covariance (which is denoted on the y-axis as "Bayesian linear regression" for this plot)—and the maximum likelihood, i.e. sigmoid probabilities, corresponding to classification of the digit 9 class. We notice fundamental differences in the representations of these relationships. Predictive entropy is represented in a desirable fashion for the correctly classified cases, in that low predictive entropy is generally associated with extreme sigmoid probabilities and high predictive entropy is generally associated with intermediate sigmoid probabilities. This suggests that predictive entropy generally performs well in capturing uncertainty and thus improves prediction when we follow an abstention criterion based on predictive entropy.

By comparison, the relationship between Gaussian covariance for correctly and incorrectly classified classes is less clear, thereby suggesting that it is not a desirable metric for uncertainty.

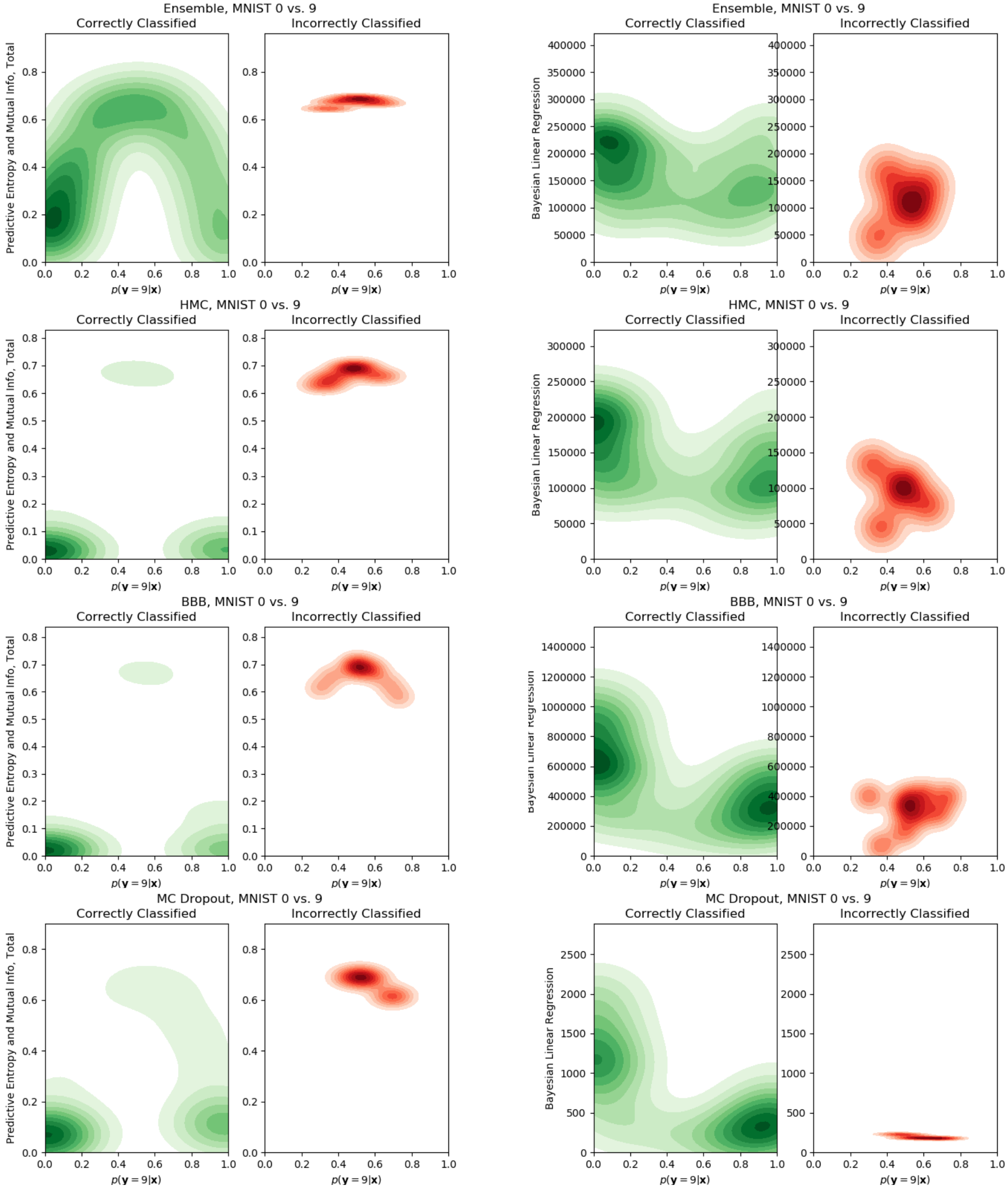


Figure 6.5 Relationship between predictive entropy and Gaussian covariance and the sigmoid probability of a given digit class output by a model, computed for all four inference methods for the conventional classification model.

Chapter Seven

Conclusions and Future Research

We organize conclusions and insights from the experiments in this thesis by regression and classification, particularly given that regression has considerably fewer uncertainty metrics than classification and several additional inference methods.

7.1 Regression

7.1.1 Uncertainty metrics

One of the novel propositions of this thesis is the application of the law of total variance decomposition (as adopted by Kendall and Gal (2017) and described in §3.3.2) to deep ensembles to estimate aleatoric and epistemic uncertainty. Per this decomposition, we compute the variance over the probabilistic NNs' estimates for the mean to quantify epistemic uncertainty, and average over the probabilistic NNs' estimates for heteroscedastic variance to quantify aleatoric uncertainty. We find that this decomposition is able to capture aleatoric and epistemic uncertainty with moderate success when applied to pedagogical synthetic regression tasks and outperforms alternatives such as MC dropout, although it suffers from the same drawbacks that plague all ensemble methods, which we discuss below underneath methods. It is also important to note that—per this model—it is impossible to obtain estimates of epistemic uncertainty that exceed those for aleatoric uncertainty at a particular x-input, which is a theoretical constraint that substantially limits the practical applications of the decomposition.

Next, we find that last layer representations may provide an attractive alternative for quantifying epistemic uncertainty compared to existing alternatives. This is in light of the fact that there are an abundance of proposed approaches for modeling per-point aleatoric variance for regression—but comparatively few suggestions for epistemic uncertainty. One metric that appears promising for obtaining reasonable estimates for epistemic uncertainty is the last layer distance between an input and the training data (§3.3.3).

However, it is important to concede that some last layer OOD metrics that map in-distribution data to zero and out-of-distribution data to non-zero values, such as Orthonormal Certificates (§3.3.3), are unable to differentiate regions with scarce training data as having some level of epistemic uncertainty. Instead, they practically treat all regions with any level of training data as possessing zero epistemic uncertainty, which deviates from the philosophical definition of epistemic uncertainty that we established at the outset of this thesis.

In general, last layer representations and OOD metrics provide an avenue for thinking about ways to estimate what is or is not in-distribution for regression, which is not a typically adopted perspective given that the goal of regression—unlike classification—is typically accepted to be generalizing beyond known regions to unknown domains, which is at odds with traditional views of OOD detection. In this sense, adapting OOD metrics from classification for regression—particularly at the level of last layer representations—may contribute a valuable novel perspective towards views on quantifying epistemic uncertainty.

7.1.2 Methods

In general, traditional heteroscedastic models for regression, including the probabilistic NN and deep ensemble, are able to produce relatively accurate estimates for aleatoric uncertainty with sufficient tuning and random restarts. That being said, it is true that the representations of uncertainty in ensembles have no theoretical guarantees due to the nature of stochastic initialization of model weights. Although not explicitly documented in the results of this thesis, it was common in our experiments for both the traditional and deep ensemble to output extreme values for uncertainty estimates, which is a considerable drawback of relying upon ensemble approaches that is well-documented in the literature.

By comparison, Bayesian heteroscedastic models (§3.2.1) are limited in their ability to capture aleatoric uncertainty, in part due to the lack of tractable inference methods for this particular model variant. We found that HMC and BBB cannot accurately approximate the posterior for the Bayesian heteroscedastic regression model, often outputting extreme values for the heteroscedastic variance, presumably to minimize the loss function in Equation 3.3.

While these shortcomings necessitate reliance upon MC dropout for the heteroscedastic uncertainty model, we find that MC dropout generally struggles to provide accurate estimates for epistemic uncertainty for both the homoscedastic and heteroscedastic noise models. This is especially so for regions with considerable in-between uncertainty, as reflected by our experiments on both the synthetic datasets and the UCI regression task. Moreover, MC dropout requires careful tuning of the dropout rate p for accurate representation of aleatoric noise, another consideration commonly documented in the literature. These concerns therefore imply that it is necessary to tread carefully when applying MC dropout to real-life regression tasks, in particular tasks with "in-between" gaps present in the data. As such, in light of these drawbacks, it is surprising to some extent that MC dropout is so heavily adopted in the literature.

Finally, based upon our synthetic regression experiments, the neural linear model appears to be an attractive approximation for HMC that can scale more readily to large data tasks and, by implication, provide a tractable way to approximate ground truth. It would therefore be desirable to more closely examine the theoretical guarantees of the regularized and MAP neural linear model, as well as the Bayesian noise neural linear (Ober and Rasmussen, 2019), which we did not consider in this work.

7.2 Classification

7.2.1 Uncertainty metrics

One novel proposition that this thesis considers is applying the principle of conditioning on model weights (which is used to estimate mutual information from predictive entropy) to decompose traditional metrics of variability for categorical distributions into their aleatoric and epistemic components. We evaluate this decomposition for several different measures of variability—including the Gini-Simpson index and so-called Allaj index—and find that it produces representations of aleatoric uncertainty and epistemic uncertainty that are consistent with the more conventional representations produced by predictive entropy and mutual information. In other words, we find high levels of aleatoric uncertainty in regions that exhibit substantial class overlap and high levels of epistemic uncertainty in regions for which it is possible to sample several different decision boundaries from the model posterior.

In fact, we find practically that these different statistical measures of variability—including the predictive entropy, Gini-Simpson index, and Allaj index—produce functionally similar, if not identical, estimates when applied to the posterior predictive of models designed for binary classification tasks. Moreover, these estimates are also similar to those provided by the law of total variance decompositions proposed by Kwon et al. (2018) and Shridhar, Laumann, and Liwicki (2018). This is so much so, actually, that the aleatoric and epistemic components of all of the aforementioned estimates are also similar, as we visualize in the results on synthetic classification.

Although this is likely an artifact that exists in part due to the limited dimensionality of binary classification tasks (and due to the fact that variability can only be expressed in so many ways for Bernoulli random variables), we also find that all of these estimates—whether we use their total, aleatoric, or epistemic components—produce identical AUC vs. retained data curves. There are two implications that arise from this result. The first is that it would be instructive to apply these estimates to multiclass classification tasks in order to better evaluate their ability to capture uncertainty. The second is that it likely would be both helpful and instructive to consider approaches for quantifying aleatoric and epistemic uncertainty that fall outside of the purview of classical statistical thinking in order to arrive at more diverse estimates for uncertainty.

While OOD metrics—including Gaussian covariance, last layer distance, largest softmax score, and functinal margin—represent one class of approaches that differ from the aforementioned statistical metrics, we find that they, in practice, do not produce better results when we use the metrics to follow an abstention criterion, as proposed by AngelosFilos, Gomez, and Rudner (n.d.). In fact, several of these metrics are wildly inconsistent and result in highly variable AUC vs. retained data curves. Given this, it is possible that such metrics might be more useful for more complex classification tasks that involve handling inputs that tend to be out of distribution rather than the simple binary tasks that we considered, for which handling data with high epistemic uncertainty was less of a concern. We leave this, as well as evaluation of other OOD metrics, for future work.

7.2.2 Methods

We find considerable differences in the expressiveness of different inference methods when constsructing and visualizing posterior predictives. In particular, we find in our synthetic classification tasks that HMC and ensemble produce posterior predictives with the most accurate visual representations of uncertainty, at least as interpreted from an intuitive human standpoint. In contrast, we find that BBB and MC dropout tend to construct narrow linear decision boundaries in their posterior predictives. While this perhaps matters from a philosophical standpoint when we consider our pedagogical synthetic classification examples, we find that the practical performance of our methods provides a different narrative.

While HMC still performs the best out of our methods when paired with predictive entropy when we compute the AUC vs. retained data curve for the MNIST binary classification task, we find that it is matched competitively by MC dropout applied to a standard classification model and even outperformed slightly by MC dropout applied to the heteroscedastic classification model proposed by Kendall and Gal (2017). Then, in contrast to its superior posterior predictive representation, ensemble underperforms on the MNIST classification. This reflects aforementioned concerns about the instability and lack of theoretical guarantees for ensemble methods.

Finally, this result also implies a need to more closely evaluate the heteroscedastic classification model and its underlying theory.

7.3 Future research

There are many promising and exciting directions for future research related to many of the topics discussed in this thesis.

First and foremost, it would be desirable to extend evaluation of all of the metrics and methods considered in this work to more real-world datasets, including more UCI benchmark regression tasks (whether with standard or gap datasets) and multiclass classification tasks. The latter is particularly important given that it is possible that many of the statistical measures of variability and metrics that we considered may be more expressive and distinct on multi-class distributions.

Related to this, it would be both interesting and valuable to evaluate how OOD detection metrics respond to inputs that are synthetically altered to correspond to human notions of out-of-distribution for classification tasks. One simple example of this is rotating a digit in the MNIST dataset to be upside down, which might be considered a type of OOD.

Another general research direction is evaluating the use of different metrics in active learning and reinforcement learning. One simple example of an active learning approach is computing and comparing the number of acquisition steps to achieve a model error of 5% on MNIST using any of the uncertainty metrics metrics that we considered as acquisition functions.

Next, we considered the use of several OOD detection metrics and last layer representations of epistemic uncertainty for both regression and classification tasks in this thesis, but there are many other metrics that have not been considered nor evaluated. Some other metrics, as documented and benchmarked by Tagasovska and Lopez-Paz (2019), include computing uncertainty as the distance from a sphere, as the distance from the linearized decision boundary (for classification), with linear random network distillation, with PCA, with a Deep Support Vector Data Description (DSVDD) model, and by training an oracle.

Furthermore, extending some of these OOD detection approaches (which are typically reserved for classification) to regression represents an exciting research direction and an opportunity to bridge traditional representations of uncertainty in regression and literature on OOD detection.

Finally, in light of the shortcomings of probability theory that we discussed in the introduction, it would be interesting to extend to deep learning one novel decomposition of aleatoric and epistemic uncertainty that is rooted in a decision-theoretic framework known as *fuzzy preference modeling*, in which aleatoric and epistemic uncertainty are computed using an approach that evaluates the plausibility of events (Senge et al., 2014; Nguyen, Destercke, and Hüllermeier, 2019). This approach combines concepts from a generalized form of version space learning and Bayesian inference, thereby providing a bridge to represent epistemic and aleatoric uncertainty distinctly.

References

Allaj, Erindi (2018). “Two simple measures of variability for categorical data”. In: *Journal of Applied Statistics* 45.8, pp. 1497–1516.

AngelosFilos, SebastianFarquhar, AidanN Gomez, and TimG J Rudner. “Benchmarking Bayesian Deep Learning with Diabetic Retinopathy Diagnosis”. In:

Barber, David and Christopher M Bishop (1998). “Ensemble learning in Bayesian neural networks”. In: *Nato ASI Series F Computer and Systems Sciences* 168, pp. 215–238.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Blundell, Charles et al. (2015). “Weight uncertainty in neural networks”. In: *arXiv preprint arXiv:1505.05424*.

Bui, Thang et al. (2016). “Deep Gaussian processes for regression using approximate expectation propagation”. In: *International conference on machine learning*, pp. 1472–1481.

Chen, Tianqi, Emily Fox, and Carlos Guestrin (2014). “Stochastic gradient hamiltonian monte carlo”. In: *International conference on machine learning*, pp. 1683–1691.

Cybenko, George (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Damianou, Andreas and Neil Lawrence (2013). “Deep gaussian processes”. In: *Artificial Intelligence and Statistics*, pp. 207–215.

Depeweg, Stefan et al. (2016). “Learning and policy search in stochastic dynamical systems with bayesian neural networks”. In: *arXiv preprint arXiv:1605.07127*.

— (2017). “Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning”. In: *arXiv preprint arXiv:1710.07283*.

Der Kiureghian, Armen and Ove Ditlevsen (2009). “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2, pp. 105–112.

Dua, Dheeru and Casey Graff (2017). *UCI machine learning repository*.

Dubois, Didier and Henri Prade (1988). “Default reasoning and possibility theory”. In: *Artificial Intelligence* 35.2, pp. 243–257.

Foong, Andrew YK et al. (2019). "'In-Between'Uncertainty in Bayesian Neural Networks". In: *arXiv preprint arXiv:1906.11537*.

Fox, Craig R and Gülden Ülkümen (2011). "Distinguishing two dimensions of uncertainty". In: *Perspectives on thinking, judging, and decision making*, pp. 21–35.

Freeman, Lintin C (1965). "Elementary applied statistics: for studies in behavioraal science/by Linton C. Freeman". In:

Gal, Yarin (2016). "Uncertainty in deep learning". In: *University of Cambridge* 1, p. 3.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*, pp. 1050–1059.

Gigerenzer, Gerd (1994). "Why the distinction between single-event probabilities and frequencies is important for psychology (and vice versa)". In: *Subjective probability*. Wiley, pp. 129–161.

Gini, Corrado (1912). "Variabilità e mutabilità". In: *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian, Jean Pouget-Abadie, et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Graves, Alex (2011). "Practical variational inference for neural networks". In: *Advances in neural information processing systems*, pp. 2348–2356.

Hafner, Danijar et al. (2018). "Reliable uncertainty estimates in deep neural networks using noise contrastive priors". In:

Hansen, Lars Kai and Peter Salamon (1990). "Neural network ensembles". In: *IEEE transactions on pattern analysis and machine intelligence* 12.10, pp. 993–1001.

Hernández-Lobato, José Miguel and Ryan Adams (2015). "Probabilistic backpropagation for scalable learning of bayesian neural networks". In: *International Conference on Machine Learning*, pp. 1861–1869.

Hinton, Geoffrey E and Drew Van Camp (1993). "Keeping the neural networks simple by minimizing the description length of the weights". In: *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13.

Houlsby, Neil et al. (2011). "Bayesian active learning for classification and preference learning". In: *arXiv preprint arXiv:1112.5745*.

Howell, William C and Sarah A Burnett (1978). “Uncertainty measurement: A cognitive taxonomy”. In: *Organizational Behavior and Human Performance* 22.1, pp. 45–68.

Hüllermeier, Eyke and Willem Waegeman (2019). “Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction”. In: *arXiv preprint arXiv:1910.09457*.

Kahneman, Daniel and Amos Tversky (1982). “Variants of uncertainty”. In: *Cognition* 11.2, pp. 143–157.

Kendall, Alex and Yarin Gal (2017). “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems*, pp. 5574–5584.

Kingma, Diederik P and Max Welling (2013). “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114*.

Kullback, Solomon and Richard A Leibler (1951). “On information and sufficiency”. In: *The annals of mathematical statistics* 22.1, pp. 79–86.

Kwon, Yongchan et al. (2018). “Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation”. In:

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in neural information processing systems*, pp. 6402–6413.

Laves, Max-Heinrich et al. (2019). “Quantifying the uncertainty of deep learning-based computer-aided diagnosis for patient safety”. In: *Current Directions in Biomedical Engineering* 5.1, pp. 223–226.

LeCun, Yann, Corinna Cortes, and Christopher JC Burges (1998). “The MNIST database of handwritten digits, 1998”. In: *URL http://yann. lecun. com/exdb/mnist* 10, p. 34.

Louizos, Christos and Max Welling (2016). “Structured and efficient variational deep learning with matrix gaussian posteriors”. In: *International Conference on Machine Learning*, pp. 1708–1716.

— (2017). “Multiplicative normalizing flows for variational bayesian neural networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2218–2227.

Malinin, Andrey and Mark Gales (2018). “Predictive uncertainty estimation via prior networks”. In: *Advances in Neural Information Processing Systems*, pp. 7047–7058.

Moberg, John (2019). “Uncertainty-aware models for deep reinforcement learning”. In:

Mohamed, Shakir and Balaji Lakshminarayanan (2016). “Learning in implicit generative models”. In: *arXiv preprint arXiv:1610.03483*.

Mukhoti, Jishnu, Pontus Stenetorp, and Yarin Gal (2018). "On the importance of strong baselines in bayesian deep learning". In: *arXiv preprint arXiv:1811.09385*.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.

Neal, Radford M et al. (2011). "MCMC using Hamiltonian dynamics". In: *Handbook of markov chain monte carlo* 2.11, p. 2.

Neal, Radford M (2012). *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.

Nguyen, Vu-Linh, Sébastien Destercke, and Eyke Hüllermeier (2019). "Epistemic uncertainty sampling". In: *International Conference on Discovery Science*. Springer, pp. 72–86.

Ober, Sebastian W and Carl Edward Rasmussen (2019). "Benchmarking the Neural Linear Model for Regression". In: *arXiv preprint arXiv:1912.08416*.

O'Hagan, Tony (2004). "Dicing with the unknown". In: *Significance* 1.3, pp. 132–133.

Osband, Ian (2016). "Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout". In: *NIPS Workshop on Bayesian Deep Learning*. Vol. 192.

Osband, Ian, John Aslanides, and Albin Cassirer (2018). "Randomized prior functions for deep reinforcement learning". In: *Advances in Neural Information Processing Systems*, pp. 8617–8629.

Osband, Ian, Charles Blundell, et al. (2016). "Deep exploration via bootstrapped DQN". In: *Advances in neural information processing systems*, pp. 4026–4034.

Pawlowski, Nick et al. (2017). "Implicit weight uncertainty in neural networks". In: *arXiv preprint arXiv:1711.01297*.

Pearce, Tim, Mohamed Zaki, Alexandra Brintrup, Nicolas Anastassacos, et al. (2018). "Uncertainty in neural networks: Bayesian ensembling". In: *arXiv preprint arXiv:1810.05546*.

Pearce, Tim, Mohamed Zaki, Alexandra Brintrup, and Andy Neely (2018). "High-quality prediction intervals for deep learning: A distribution-free, ensembled approach". In: *arXiv preprint arXiv:1802.07167*.

Peterson, Dane K and Gordon F Pitz (1988). "Confidence, uncertainty, and the use of information." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 14.1, p. 85.

Pinsler, Robert et al. (2019). "Bayesian batch active learning as sparse subset approximation". In: *Advances in Neural Information Processing Systems*, pp. 6356–6367.

Riquelme, Carlos, George Tucker, and Jasper Snoek (2018). "Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling". In: *arXiv preprint arXiv:1802.09127*.

Senge, Robin et al. (2014). "Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty". In: *Information Sciences* 255, pp. 16–29.

Shafer, Glenn (1976). *A mathematical theory of evidence.* Vol. 42. Princeton university press.

Shannon, Claude E (1948). "A mathematical theory of communication". In: *Bell system technical journal* 27.3, pp. 379–423.

Shridhar, Kumar, Felix Laumann, and Marcus Liwicki (2018). "Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference". In: *arXiv preprint arXiv:1806.05978*.

Simpson, Edward H (1949). "Measurement of diversity". In: *nature* 163.4148, pp. 688–688.

Skorokhodov, Ivan and Mikhail Burtsev (2019). "Loss surface sightseeing by multi-point optimization". In: *arXiv preprint arXiv:1910.03867*.

Snoek, Jasper et al. (2015). "Scalable bayesian optimization using deep neural networks". In: *International conference on machine learning*, pp. 2171–2180.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Sun, Shengyang et al. (2019). "Functional variational bayesian neural networks". In: *arXiv preprint arXiv:1903.05779*.

Tagasovska, Natasa and David Lopez-Paz (2018). "Frequentist uncertainty estimates for deep learning". In: *arXiv preprint arXiv:1811.00908*.

— (2019). "Single-Model Uncertainties for Deep Learning". In: *Advances in Neural Information Processing Systems*, pp. 6414–6425.

Tanno, Ryutaro et al. (2019). "Uncertainty Quantification in Deep Learning for Safer Neuroimage Enhancement". In: *arXiv preprint arXiv:1907.13418*.

Taylor, James W (2000). "A quantile regression neural network approach to estimating the conditional density of multiperiod returns". In: *Journal of Forecasting* 19.4, pp. 299–311.

Volz, Kirsten G, Ricarda I Schubotz, and D Yves von Cramon (2005). "Variants of uncertainty in decision-making and their neural correlates". In: *Brain research bulletin* 67.5, pp. 403–412.

Walley, Peter (1991). *Statistical reasoning with imprecise probabilities. 1991.*

Welling, Max and Yee W Teh (2011). "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688.

White, Halbert (1992). “Nonparametric estimation of conditional quantiles using neural networks”. In: *Computing Science and Statistics*. Springer, pp. 190–199.

Yao, Jiayu et al. (2019). “Quality of uncertainty quantification for Bayesian neural network inference”. In: *arXiv preprint arXiv:1906.09686*.

Zhang, Guodong et al. (2017). “Noisy natural gradient as variational inference”. In: *arXiv preprint arXiv:1712.02390*.

Zhang, Ruqi et al. (2019). “Cyclical stochastic gradient mcmc for bayesian deep learning”. In: *arXiv preprint arXiv:1902.03932*.

Appendix A

Hyperparameter Settings

A.1 Neural networks and ensembles

We use a learning rate of 10^{-2} for all deterministic and probabilistic NNs for synthetic regression and classification tasks and 10^{-4} for MNIST classification, with a regularization rate λ of 0.01 for all experiments. Weights are randomly initialized for each network.

We construct both our ensembles and deep ensembles with 30 NNs and probabilistic NNs each, respectively. We bootstrap the NNs for the traditional ensembles but not for the deep ensembles, following the recommendation of Lakshminarayanan, Pritzel, and Blundell (2017) to let each network in the deep ensemble train on the full set of randomly shuffled training data.

A.2 Hamiltonian Monte Carlo

We sample the momentum variable in HMC from $\mathcal{N}(0, \mathbf{I})$ and use 30 leapfrog steps, with an initial stepsize η of 5×10^{-3}. We check acceptance rate every 100 iterations and decrease it by a factor of 0.9 if it is less than 0.5 and increase it by a factor of 1.14 if it is greater than 0.8. We use a burn-in of 0.3 of the total number of iterations and a thinning factor of 2.

A.3 Bayes by Backprop

We use a learning rate of 10^{-3} for all classification tasks, 5×10^{-4} for Datasets 1 and 3 and 10^{-3} for Dataset 2 for synthetic regression.

A.4 Monte Carlo dropout

After performing an extensive gridsearch over different dropout rates, we choose to set the dropout rates p to $0.2, 0.3, 0.4$ for standard MC dropout and $0.1, 0.3, 0.4$ for the LOTV decomposition of MC dropout for Datasets 1, 2, and 3, respectively, in synthetic regression.

We use a dropout rate of 0.02 for UCI regression and 0.3 for all synthetic and real classification tasks.

We set the learning rate to 10^{-4} for all synthetic regression tasks, 10^{-2} for all classification tasks, and 10^{-3} for UCI regression. The regularization rate λ is 0.01 for all tasks.

A.5 Neural linear

We use a learning rate of 10^{-3} for all synthetic regression tasks for both the MAP and regularized neural linear models.

Appendix B

Additional Results

B.1 Synthetic regression

We include visual results that were not reviewed in §5.1.2.

B.2 Synthetic classification

We include visual results that were not reviewed in §5.2.2.

B.3 MNIST classification

We include visual results that were not reviewed in §6.2.3.

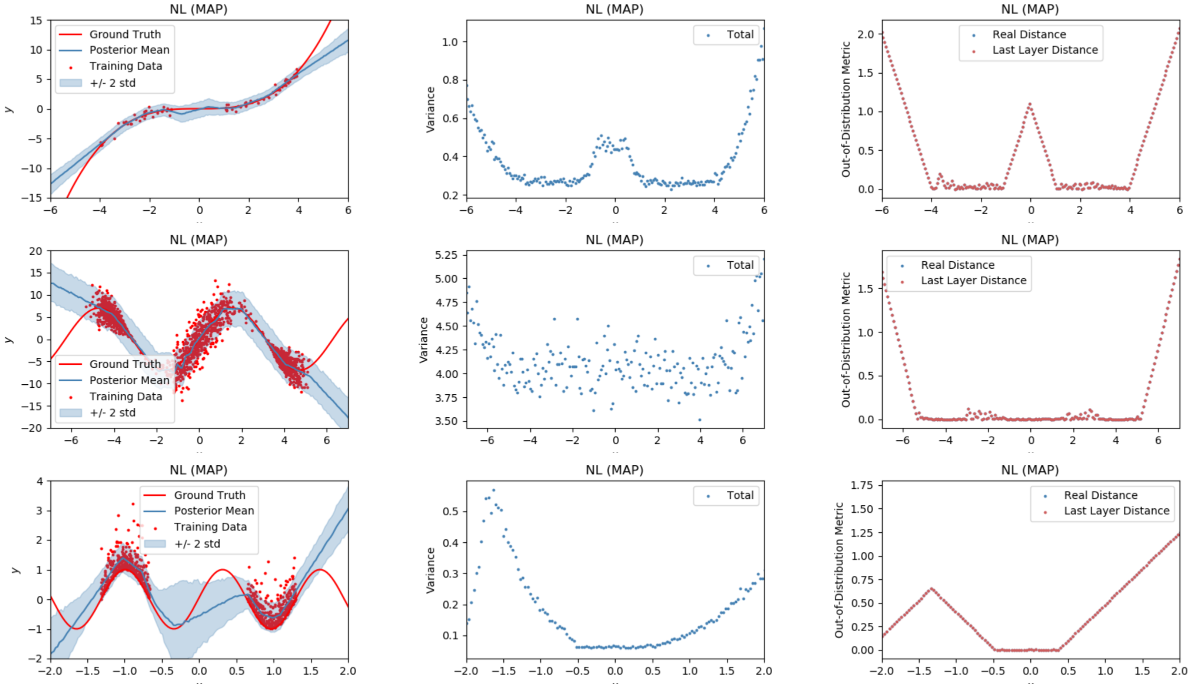


Figure B.1 A comparison of the results from the MAP NL method for regression on Datasets 1, 2, and 3.

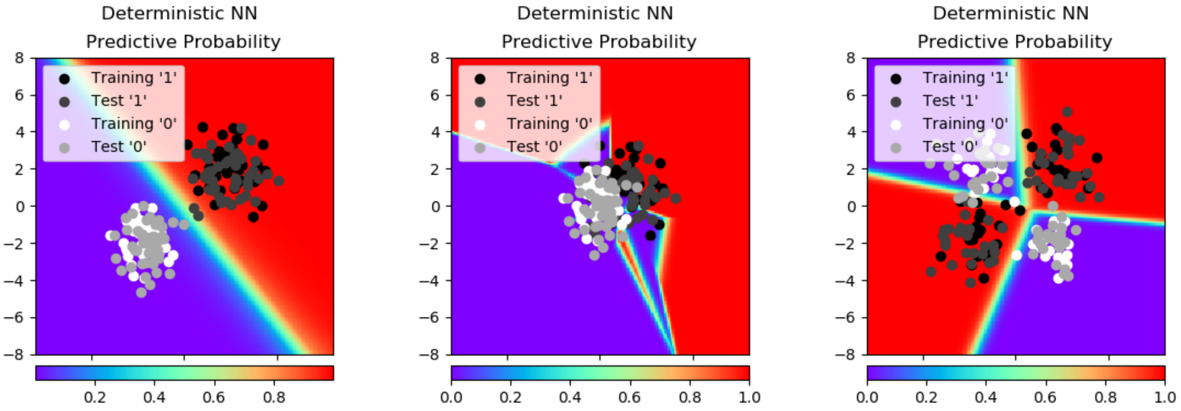


Figure B.2 The predictive probability of a deterministic NN for classification on Datasets 1, 2, and 3.

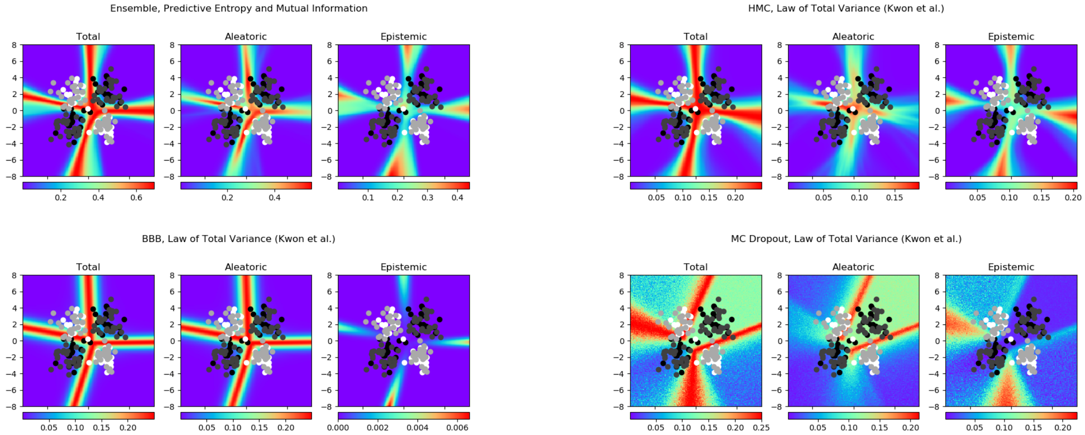


Figure B.3 LOTV as proposed by Kwon et al. (2018) applied to the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 2.

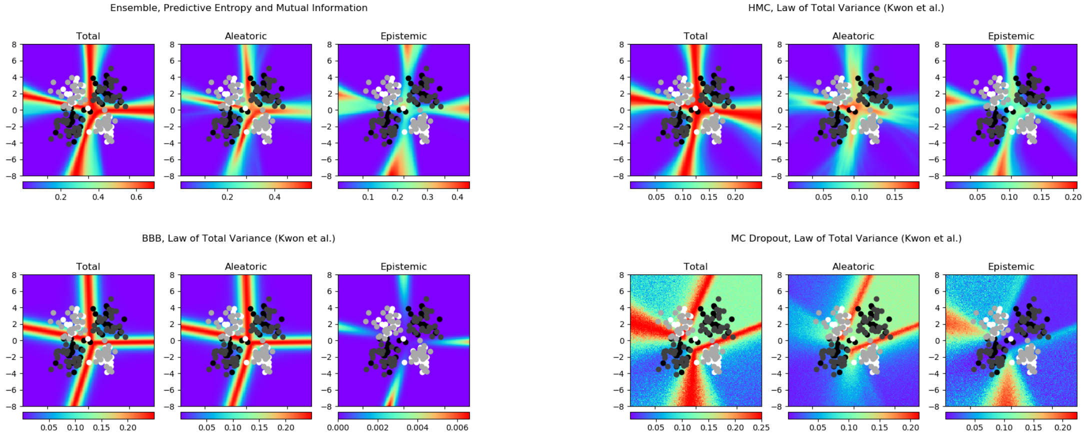
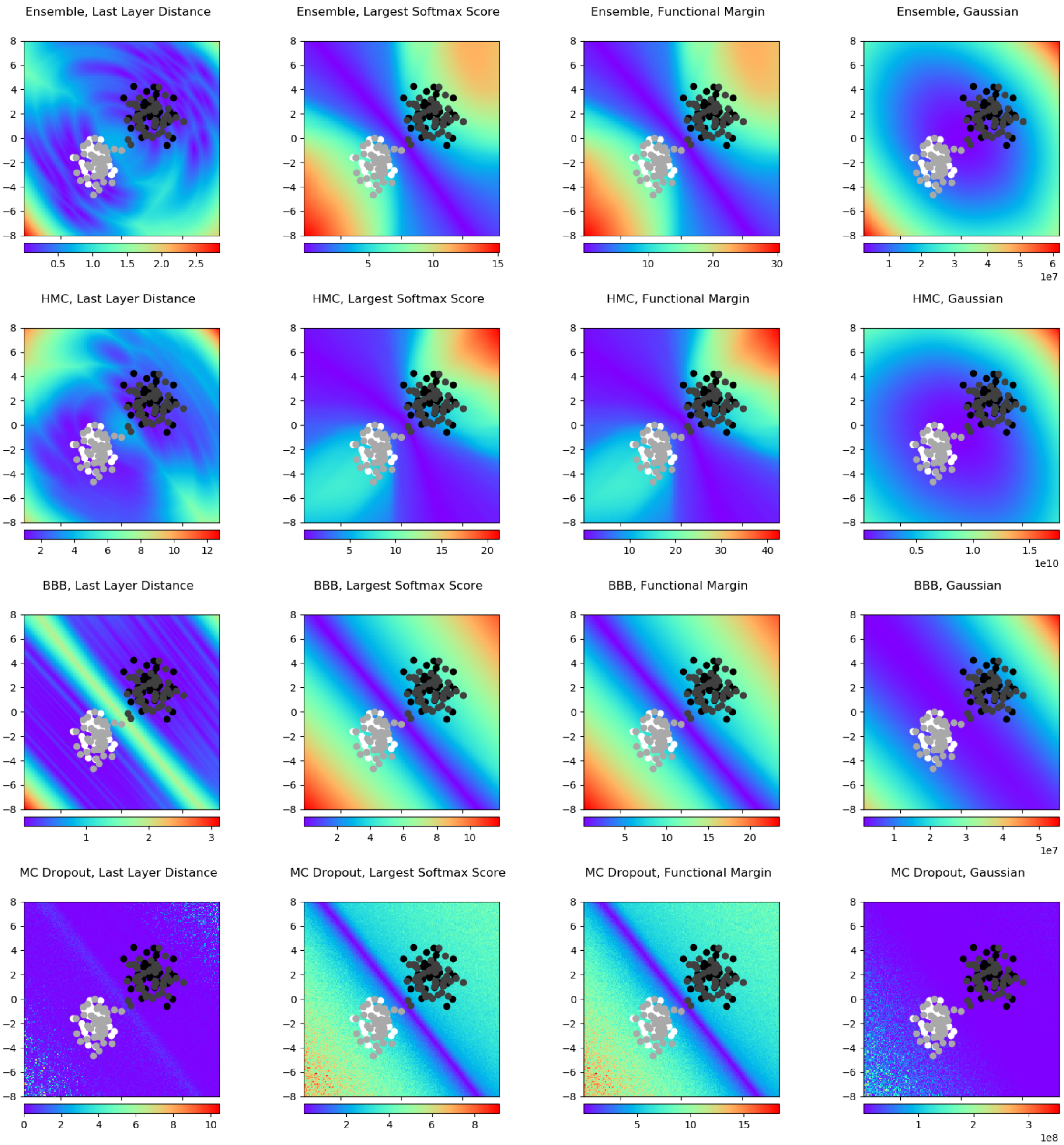


Figure B.4 LOTV as proposed by Kwon et al. (2018) applied to the posterior predictive of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 3.


Ensemble, Last Layer Distance

Ensemble, Largest Softmax Score

Ensemble, Functional Margin

Ensemble, Gaussian

HMC, Last Layer Distance

HMC, Largest Softmax Score

HMC, Functional Margin

HMC, Gaussian

BBB, Last Layer Distance

BBB, Largest Softmax Score

BBB, Functional Margin

BBB, Gaussian

MC Dropout, Last Layer Distance

MC Dropout, Largest Softmax Score

MC Dropout, Functional Margin

MC Dropout, Gaussian

Figure B.5 OOD metrics applied to the last layer (last layer distance, Gaussian) and output layer (largest softmax score, functional margin) representations of the conventional classification model, with ensemble, HMC, BBB, and MC dropout as inference methods, for Dataset 1.

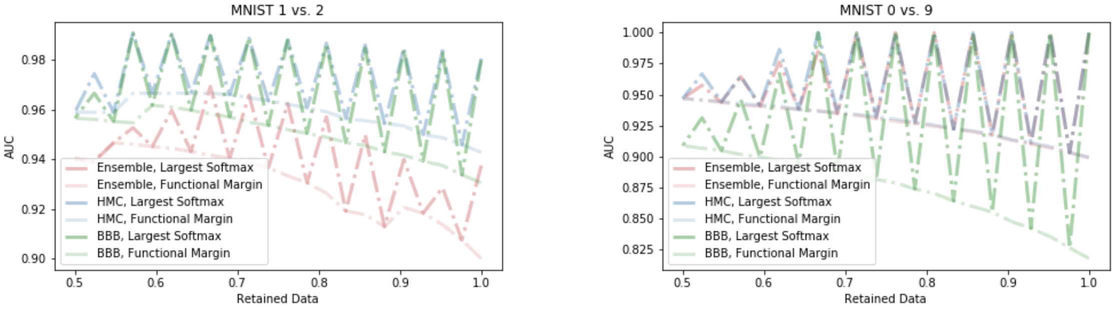


Figure B.6 AUC vs. retained data based upon three OOD detection metrics applied to the conventional classification model, with ensemble, HMC, and BBB as the inference methods, for MNIST Datasets 1 and 2.