Carnegie Mellon University

Uncertainty Quantification in Simulation-based Inference

A Thesis Presented for

The Doctor of Philosophy

in

Statistics

by

Niccolò Dalmasso

Department of Statistics & Data Science
Carnegie Mellon University
Pittsburgh, PA 15213

May 2021

Dedicated to my parents and my brother.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor Ann Lee for her guidance and tireless support throughout my PhD, as well as having shared countless productive research discussions. I am especially grateful for her patience, kindness and empathy when I was struggling, either with research or personal life, as well as having guided me towards graduation during a global pandemic. Without her help and support, this dissertation would not have been possible.

I would like to deeply thank Rafael Izbicki for being an amazing collaborator, a patient mentor, and having shared many intuitions and insights that have made me a better researcher. I would also like to extend my thanks to Mikael Kuusela, Peter Freeman, Larry Wasserman and Cosma Shalizi for being on my committee and their valuable suggestions. A special thanks to Mikael, who has provided me invaluable feedback for high-energy physics applications, and with whom I have shared the "behind the scenes" of the statistical methods for the physical sciences (STAMPS) research group. Being part of STAMPS has been a tremendous learning experience and I am deeply indebted to Ann, Mikael and Larry for having allowed me to take a front-row seat. I am thankful to Robin Mejia and Alessandro Rinaldo for their support and constant encouragement throughout my PhD and their guidance in navigating hard and uncertain times during my first two years in the department. I would like to acknowledge Chad Schafer and Alex Reinhart for their collaborative efforts and insightful research interactions. I also would like to thank Aaditya Ramdas for the extensive feedback provided to me at my STAMPS talks, Isabella Verdinelli for all the encouraging words and Rebecca Nugent for constant support I first walked into the department. My gratitude also goes to Alastair Young, David Van Dyk, Roberta Sirovich and Daniel Mortlock for having supported my PhD applications and allowed me to reach this milestone in the first place.

I want to take this opportunity to thank all the mentors who have played a crucial role throughout my life: Mario Tretola, who encouraged an 11 years-old me to pursue my curiosity for Mathematics; David Carvalho Salazar, who showed me that hard work beats talent when talent doesn't work hard enough; Gianni Longhi, who supported and guided me in choosing an undergraduate degree; Roberta Sirovich, who introduced me to Statistics in the first place; Pierferdinando Rossi, who was an incredible research role model at the Scuola of Studi Superiori and supported me in continuing my academic career abroad. Finally, I am forever indebted to my grandfather, Natale Zumbo, whose kindness and empathy continue to be a guiding light; I wish you could be here to see this.

My gratitude also goes to my longtime friends: Riccardo, my brother from another mother; Andrea, a caring friend I can always count on; Federica, who bears with me since my high-school days, as well as Beatrice, Alessandro, and my friends from Cuneo; Mattia Boni, who has never forgotten the 50KHG bond; Mattia Cerri, for being always present; Matteo and Lorenzo, whose friendship and encouragements helped me stay afloat during my PhD, as well as Danilo, Federico, Alessandro and Giuseppe for many dinners and card games at Turin and Imperia; and Sara, who offered a helping hand in a tough time without thinking twice.

This acknowledgment section would not be complete without sincerely thank all my friends and colleagues I have shared these last five years with. I would like to thank Alan, my classmate, collaborator, office mate, and most importantly friend, with whom the PhD life was much more enjoyable. I also want to extend my thanks to Kwangho, Ben and Robin, for all the memories, support and work together over the past five years. I am grateful to the Italian crew at CMU as well as outside: Francesca, for being a supportive friend and for our endless enjoyable office chats; Giuseppe, for showing me the ropes in the department; Matteo, with whom I have shared many laughs and great moments playing soccer, boxing and working out; Luca, for all the memories and the chats since our IBM time together; Luca and Alessandro, for sharing a laugh, a coffee or a soccer match. I am also grateful to Shamindra for being a thoughtful and caring friend. I would like to acknowledge the colleagues I've had the pleasure to collaborate with during my time in the department: Ilmun, Taylor, Yotam, David, Trey and Galen, as well as the ones already mentioned above. A special thanks go to those friends who helped me keep my sanity when the global pandemic was having the better of me: Mikaela, Nil-Jana, Holly, Sasha and Nick. A shoutout also to my friends and golf buddies Alec and Ron. I would like to thank many other people, including, but not limited to, Jisu, Jerzy, Nick, Lee, Collin, Purvasha, Brendan, Jining, Ciaran, Maria, Yufei, Tudor, Lorenzo, Michael, Ian and Luca.

Last but not least, I owe the biggest thanks of all goes to my partner Cecilia, whose love and support throughout the last five years have made all of this possible. Words cannot express how fortunate and grateful I feel to have you by my side.

Questa tesi è dedicata a mio padre, a mia madre e a mio fratello. I vostri sacrifici, il vostro supporto e amore incondizionato, l'affrontare con forza i silenzi e le difficoltà di avere un figlio o un fratello lontano da casa, sono per me fonte di ispirazione e di orgoglio quotidiano. Non riesco ad immaginare come avrei potuto farcela senza di voi.

Hunc igitur terrorem animi tenebrasque necessest
non radii solis neque lucida tela diei
discutiant, sed naturae species ratioque.
Lucretius, De Rerum Natura (I, 146-148)

Please remember: things are not what they seem.
Haruki Murakami, 1Q84

Invece padron 'Ntoni aveva fatto quel viaggio lontano, più lontano di Trieste e d'Alessandria d'Egitto, dal quale non si ritorna più; e quando il suo nome cadeva nel discorso, mentre si riposavano, tirando il conto della settimana e facendo i disegni per l'avvenire, all'ombra del nespolo e colle scodelle fra le ginocchia, le chiacchiere morivano di botto, che a tutti pareva d'avere il povero vecchio davanti agli occhi, come l'avevano visto l'ultima volta che erano andati a trovarlo in quella gran cameraccia coi letti in fila, che bisognava cercarlo per trovarlo, e il nonno li aspettava come un'anima del purgatorio, cogli occhi alla porta, sebbene non ci vedesse quasi, e li andava toccando, per accertarsi che erano loro, e poi non diceva più nulla, mentre gli si vedeva in faccia che aveva tante cose da dire, e spezzava il cuore con quella pena che gli si leggeva in faccia e non la poteva dire. Quando gli narrarono poi che avevano riscattata la casa del nespolo, e volevano portarselo a Trezza di nuovo, rispose di sì, e di sì, cogli occhi, che gli tornavano a luccicare, e quasi faceva la bocca a riso, quel riso della gente che non ride più, o che ride per l'ultima volta, e vi rimane fitto nel cuore come un coltello. Così successe ai Malavoglia quando il lunedì tornarono col carro di compar Alfio per riprendersi il nonno, e non lo trovarono più.
Giovanni Verga, I Malavoglia (Chp. XV)

Abstract

Complex phenomena in engineering and the sciences are often modeled by feed-forward simulators that implicitly encode the likelihood function. Classical methods for constructing confidence sets and hypothesis tests are poorly suited for such settings. While the field of simulation-based inference has undergone a revolution in terms of the complexity of problems that can be tackled, the development on the statistical methodology front has fallen behind. Indeed, many techniques have been developed for learning a surrogate likelihood using forward-simulated data, but these methods do not guarantee frequentist confidence sets and tests with nominal coverage and Type I error control, respectively, outside the asymptotic and low-dimensional regimes. In this thesis we introduce a series of statistical tools for uncertainty quantification in a simulation-based inference setting.

In the first part of the thesis, we provide inferential tools with frequentist guarantees for a high-dimensional simulator-based setting. We introduce a statistical framework that unifies classical statistics with modern machine learning algorithms to achieve the following goals: (i) confidence sets and hypothesis tests with finite-sample guarantees of coverage and power, (ii) diagnostics for checking empirical coverage over the entire parameter space and (iii) scalable and modular procedure which can also be used with other simulation-based approaches. We showcase the applicability of this framework across a diverse range of data and parameter settings.

In the second part of the thesis, we consider the problem of assessing the quality of fit of approximate likelihood models. Approximate likelihood models are used in settings with high-resolution and computationally intensive simulators for faster inference. We propose a statistically consistent validation method that can pinpoint the locations in the parameter space where the fit is inadequate as well as provide insights as to how the high-resolution simulator and approximate likelihood model differ.

In the third and final part of the thesis, we propose a new flexible method for nonparametric conditional density estimation that can transform any neural network regression architecture into a conditional density estimator. We demonstrate the versatility of our approach for two applications with convolutional and recurrent neural networks, respectively.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Stochastic numerical simulators have become an essential tool to study complex phenomena in many scientific domains. Simulations encode and chain together the underlying physical or biological processes to generate synthetic realizations of such phenomena. Stochastic simulators work in a feed-forward fashion: given input parameters θ, both deterministic and stochastic mechanisms are combined to produce synthetic data $\mathbf{x}$. Examples of stochastic forward simulators span the entire scientific spectrum at many different scales, including high-energy physics (Sjöstrand et al., 2001), neuroscience (Lee et al., 2006), epidemiology (Edlund et al., 2010), biology (Vázquez et al., 2003), climate science (Hurrell et al., 2013) and cosmology (Smith et al., 2003) among others. One could regard forward stochastic simulators as "black boxes" that implicitly encode the likelihood describing the relationship between parameters (the input) and observable data (the output). Indeed, the generative process of complex phenomena is often better understood and easier to encode than positing a probability distribution on the simulated data as a function of θ. Statistical inference in such setting, i.e., determining which parameters are consistent with the observed data without an explicit likelihood, is known as *simulation-based inference* (SBI) or *likelihood-free inference* (LFI). Although the two terms are sometimes used synonymously in the literature, in this thesis we use SBI as an umbrella term for inference methods relying on a simulator, while LFI refers to settings in which the simulator is intrinsically stochastic (i.e., for which the encoded processes are stochastic in nature) and that can be queried on the fly without a significant computational overhead. As we note in Section 1.3, the LFI setting also extends to situations in which a stochastic emulator is fit to a high-fidelity computationally intensive simulator to speed up probabilistic modelling.

While the field of simulation-based inference has undergone a revolution in terms of the complexity of problems that can be tackled (see Cranmer et al. (2020) for a recent review), the development on the statistical methodology front has fallen behind. Overall, most of the attention in the likelihood-free inference literature has been directed to making inference algorithms efficient and scalable to high-dimensional data, and considerably less so to the frequentist statistical properties of the resulting downstream inference. This thesis aims to provide an arsenal of practical and flexible inferential tools that can be used not only to construct inference with frequentist guarantees on validity and power, but also to complement other frequentist LFI approaches which focus on estimating key quantities like the likelihood or likelihood ratio function. In general, a recurring theme within this thesis is marrying the high-capacity and computational efficiency brought by machine learning algorithms with the benefits of the cornerstones of classical statistics, such as hypothesis testing and confidence sets (Chapter 2), two-sample tests (Chapter 3) and conditional density estimation (Chapter 4).

1.1 Traditional Simulation-based Inference Approaches

Broadly speaking, the goal of LFI is to compare simulated and observed data to identify parameter values compatible with the observed data. However, a direct comparison is not always viable, especially when simulators often output high-dimensional or non-standard data such as images and time-series. Traditional approaches hence rely on defining a set of summary statistics for such comparison, which reduces the dimensionality of the problem and allows one to incorporate domain scientific knowledge. The first example of deriving scientifically informed summary statistics was proposed by Diggle and Gratton (1984) for particle collisions experiments in high-energy physics. Their proposed scheme, consisting of estimating the distribution of 1D photon counts via nonparametric histogram estimators, would later become a key component in the discovery of the Higgs Boson (Aad et al., 2012).

The best known traditional LFI method is perhaps *Approximate Bayesian Computation* (ABC). Loosely speaking, ABC methods provide a sample from the posterior distribution by first sampling from the parameter space and then rejecting all parameter values for which the simulated data and observed data summary statistics are not sufficiently close. The notion of "closeness" is defined based on a distance metric between summary statistics (usually Euclidean) and a tolerance level. The latter provides a direct trade-off in terms of accuracy and efficiency; smaller tolerance levels correspond to better posterior approximations but longer computing times, as one needs to generate simulations which are close enough to the observed data, and vice versa. The literature on ABC is vast and has resulted in many successful applications, including in genetics (Beaumont and Rannala, 2004), biology (Ratmann et al., 2007), ecology (Wood, 2010), economics (Peters and Sisson, 2006) and cosmology (Lewis and Bridle, 2002); we refer the reader to Sisson et al. (2018), Karabatsos and Leisen (2018) and Beaumont (2019) for extensive reviews.

Traditional frequentist LFI approaches rely on repeatedly querying the simulator, such as *Monte Carlo* (MC) and *Indirect Inference* (II) methods. MC approaches construct the distribution of summary statistics across the parameter space by repeated simulations at each parameter value (Barlow and Beeston, 1993; Weinzierl, 2000; Schafer and Stark, 2009). II methods do not rely on summary statistics but instead fit an auxiliary model (i.e., a simple and possibly misspecified model) on the simulated data. For each simulation, they record the correspondence between the parameters of the simulator and the parameters of the auxiliary model. Inference is drawn by selecting the parameters associated with the auxiliary model parameter values for the observed data over multiple simulation rounds (Gourieroux et al., 1993).

1.2 Recent Developments in Likelihood-Free Inference

In recent years, LFI methods have taken advantage of the increase in efficiency and capacity brought by machine learning algorithms and modern computing power. This has resulted in approaches that require fewer simulations and that can accommodate high-dimensional data directly without the use of summary statistics or auxiliary models. Meaningful scientific contributions of such methods include high-energy physics (Brehmer et al., 2018), cosmology (Alsing et al., 2019) and neuroscience (Gonçalves et al., 2019) among others. Recent LFI approaches can be roughly grouped based on whether they estimate (i) the posterior distribution, (ii) the likelihood or (iii) density ratios such as the likelihood over the marginal or likelihood ratios.

Methods that estimate the posterior distribution directly include the use of random forest (Marin et al., 2016), Gaussian copula (Chen and Gutmann, 2019), nonparametric conditional density estimators (Izbicki et al., 2019), neural networks (Radev et al., 2020) and neural density estimators (sequential neural posterior estimation, SNPE; Papamakarios and Murray 2016; Lueckmann et al. 2017; Greenberg et al. 2019). Comparison with traditional ABC methods has shown that SNPEs approaches can achieve the same accuracy in posterior inference with orders of magnitude fewer simulations (Papamakarios et al., 2021).

Among approaches that focus on estimating the likelihood, a popular approach is to approximate the likelihood via Gaussian synthetic surrogates (Wood, 2010; Meeds and Welling, 2014; Wilkinson, 2014; Price et al., 2018; Fasiolo et al., 2018; Picchini et al., 2020), which have been combined with active learning to minimize the simulation costs (Gutmann et al., 2016; Järvenpää et al., 2019, 2021). Other approaches include the use of neural density estimators (Lueckmann et al., 2019; Papamakarios et al., 2019; Wiqvist et al., 2021) and variational inference techniques (Ong et al., 2017; M. Tran et al., 2017; Louppe et al., 2019).

Finally, a separate stream of work has focused on density ratio estimation, with the first nonparametric estimator provided by Izbicki et al. (2014). Most density ratio estimation methods leverage the fact that density ratios can be learned via binary classification (Sugiyama et al., 2012; Mohamed and Lakshminarayanan, 2016). In Bayesian inference, posterior estimation can be achieved by approximating of the ratio between the likelihood ratio and the marginal, using logistic regression Thomas et al. (2021) and neural networks (Hermans et al., 2020; Durkan et al., 2020). In frequentist inference, Cranmer et al. (2015) provide a likelihood ratio statistic estimator via binary classification, which can be made more efficient provided the joint likelihood ratio and joint score are available (Stoye et al., 2018; Brehmer et al., 2020b).

In general, the main drivers behind the increase in both capacity and efficiency in LFI approaches have been flexible non-parametric machine learning algorithms, with the most prominent example being neural conditional density estimators such as autoregressive models (Uria et al., 2014, 2016; Oord et al., 2016; Oord et al., 2016) and normalizing flows (e.g., Kingma et al. 2016 and Papamakarios et al. 2017; see Papamakarios et al. 2021 for a review). Although most of the applications showcasing the benefit of such approaches are in Bayesian inference, these are also applicable in a frequentist inference framework. Indeed, surrogates of the likelihood or likelihood ratio function can be used for frequentist inference, as in, e.g., Brehmer et al. (2020b). Notably, also the posterior can be used as a tool for frequentist inference. For instance, Neiswanger and Ramdas (2021) use the prior-posterior ratio fit by a Gaussian process to construct valid finite-sample confidence sets, while Thornton et al. (2017) show the posterior computed via ABC can provide asymptotically valid inference if a specific data-driven prior is used.

1.3 Connections to Deterministic Simulators

A further distinction within the simulation-based inference literature is based on the nature of the simulator itself. As mentioned, LFI refers to settings in which the encoded processes are stochastic in nature, such as in particle collisions experiments. However, there exists a large literature for deterministic simulators, i.e., forward operators that associate a given input $\mathbf{x}$ to a single output $\mathbf{y} = F(\mathbf{x})$. An example of this is the Community Earth System model in climate science (CESM, Hurrell et al. 2013), in which each component of the climate environment (e.g., atmosphere, land, river runoff) is modeled with a separate set of differential equations which are run forward in time for prediction. Similarly, N-body simulations in cosmology provide a high-fidelity representation of gravitational interactions between particle pairs (Kacprzak et al., 2016; Abbott et al., 2019). In such applications, characterizing the relationship between the input $\mathbf{x}$ and output $\mathbf{y}$ is known as an *ill-posed inverse problem* (O'Sullivan, 1986; Evans and Stark, 2002). The notion of "ill-posedness" comes from the fact that solutions in these settings are not stable — they do not vary continuously with the observations — making a problem that is not mathematically well-posed (i.e., in which a solution exists, is unique and stable). It is important to note that, unlike the LFI setting, there is no uncertainty connected to deterministic simulators. More specifically, using the same language as in Matthies (2007), there is no *aleatoric* uncertainty in the simulator (that is, no intrinsic statistical uncertainty). In some cases, aleatoric uncertainty is added post-hoc to reflect measurement uncertainty, that is considering $\mathbf{y} = F(\mathbf{x}) + \epsilon$, where ϵ is usually Gaussian noise.

Although ill-posed inverse problems have been approached from many different angles in the literature, the most common strategy is to regularize the problem, i.e., impose additional structure, so that a solution becomes identifiable. A potential approach to make inference feasible is to fit an approximate likelihood or emulator model to the simulated data, usually in the form of a Gaussian process (Rasmussen and Williams, 2005) where the prior distribution acts as regularizer. Emulator models introduce *epistemic* uncertainty, which is a source of error that depends on the finite amount of samples and simulation runs available; unlike the aleatoric uncertainty, epistemic uncertainty could be reduced given enough computational resources. A common approach in this setting is to design a sampling strategy for input points $\mathbf{x}$ to run to minimize the emulator model predictive variance, i.e., provide the best approximation possible with a limited budget of simulation runs (Gramacy, 2020). Another solution is to run the simulator only for a few simulations in a format of batches or ensembles, where an ensemble is a collection of multiple realizations (e.g., corresponding to different initial conditions), and train an emulator model on such ensemble runs. This method is routinely used in the climatology literature as an efficient consistency testing tool of new simulation runs (Baker et al., 2015b). Yet another approach for ill-posed inverse problems aims to characterize F by discretizing the input space and estimate a linear operator K such that $F(\mathbf{x}) = K\mathbf{x}$. Since K is usually ill-conditioned (i.e., the singular values decay quickly due to the underlying problem being ill-posed), estimates of $\mathbf{x}$ are obtained using regularization techniques such as the Tikhonov regularization (see e.g., Engl et al. (1996) and Kaipio and Somersalo (2007)). Such estimates can lead to inference from both a frequentist (Stark, 1992) and Bayesian (Stuart, 2010) perspective, and has lead to meaningful scientific contributions in scientific domains such as astrophysics (Connors et al., 2006), climate science (Rodgers, 2000; Patil et al., 2020) and medical imaging (Weir, 1997). Finally, it is important to note that the aleatoric uncertainty structure in ill-posed inverse problems is usually much simpler than the one in LFI settings, as the uncertainty does not encode the randomness of the generative process, but rather the measurement uncertainty on the output $\mathbf{y}$. For this reason, LFI approaches might not be appropriate in ill-posed inverse problem settings.

1.4 Gaps in the Literature

While most of the attention in the simulation-based inference literature has been directed towards efficiency and scalability and less so towards statistical inference, frequentist procedures have played an important role in many applications. In high energy physics, for instance, classical statistical techniques (e.g., hypothesis testing for outlier detection) have resulted in discoveries of new physics and other successful applications (Feldman and Cousins, 1998; Cranmer, 2015; Tanabashi et al., 2018; Cousins, 2018). Even though controlling type I error probabilities is important in these applications, most simulator-based methods do not have theoretical guarantees on validity or power beyond low-dimensional data settings and large sample theory assumptions (Feldman and Cousins, 1998). For example, some recent LFI methods (Frate et al., 2017; Brehmer et al., 2018, 2020b) are able to estimate likelihood functions for high-dimensional data, but these methods either assume that the log-likelihood ratio has an asymptotic chi-squared distribution, or rely on Monte Carlo samples at fixed parameter settings for inference (an approach that does not scale to computations of frequentist confidence sets).

Additionally, in LFI settings where fitting an emulator to a high-fidelity computationally intensive simulator is a key step for inference, there is no statistically consistent way of quantifying the quality of the emulator fit. Existing approach only provide an overall assessment of the quality of the fit. For instance, existing two-sample tests can only provide a binary answer of the form "reject" or "fail to reject" (Lopez-Paz and Oquab, 2017; Kim et al., 2021), and consistency checks based on the resulting posterior distribution which have been used in the LFI literature (Cook et al., 2006; Talts et al., 2018) might fail to identify clearly misspecified likelihood models as these tools were originally designed for checking posterior models. Furthermore, in cases when the emulator fit is deemed inadequate, diagnose the fit of the model is of primary importance. There are currently no tools in the LFI literature that allow to locate where in the parameter space the fit is poor and to determine the difference in a potentially high-dimensional feature space between the emulator and the high-fidelity simulator samples.

1.5 Summary of Contributions

In this thesis, we develop statistical methodologies for quantifying uncertainty in simulation-based inference and we demonstrate their applicability on a variety of settings.

- **Chapter 2** develops a machinery for frequentist inference in a LFI setting to construct and diagnose confidence sets and hypothesis testing. This novel inference machinery (i) provide confidence sets and hypothesis tests with finite-sample guarantees of coverage and power, (ii) offer rigorous diagnostics to checking empirical coverage over the entire parameter space and (iii) present scalable and modular procedure that existing simulation-based approaches can borrow;

- **Chapter 3** introduces a framework to diagnose the fit of an approximate likelihood model, which we apply to emulator models fit to high-fidelity computationally intensive simulators in a LFI setting. The proposed framework can distinguish any arbitrary misspecified model from the target likelihood, and in addition can identify with statistical confidence the regions of the parameter as well as feature space where the fit is inadequate;

- **Chapter 4** contributes a new tool to the class of nonparametric neural conditional density estimators. The proposed approach can re-purpose virtually any deep neural network regression estimator into a conditional density estimator with minimal computational overhead;

- Finally, **Chapter 5** concludes with a brief summary and highlights potential directions for future work.

1.6 Bibliographic Notes

The results in Chapter 2 are based on joint work with David Zhao, Rafael Izbicki and Ann B. Lee in Dalmasso et al. (2020a) and Dalmasso et al. (2021). Chapter 3 is based on joint work with Taylor Pospisil, Ilmun Kim, Chieh-An Lin, Rafael Izbicki and Ann B. Lee in Dalmasso et al. (2020b), and Chapter 4 is based on joint work with Taylor Pospisil, Peter E. Freeman, Alex I. Malz, Rafael Izbicki and Ann B. Lee in Dalmasso et al. (2020).

Chapter 2

Bridging Classical Statistics and Likelihood-Free Inference

Hypothesis testing and uncertainty quantification are the hallmarks of scientific inference, with a long history in statistics (Fisher, 1925; Neyman, 1935; Feldman and Cousins, 1998; Chuang and Lai, 2000). Methods that achieve good statistical performance often rely on being able to evaluate a likelihood function which relates parameters of the data-generating process to observed data. However in LFI, forward simulators define a likelihood function implicitly, meaning that the likelihood is not available analytically and cannot be evaluated. While the field of likelihood-free inference has undergone a revolution in terms of the complexity of problems that can be tackled (see Cranmer et al. 2020 for a recent review), the development on the statistical methodology front has fallen behind. Indeed, a question that has received little attention so far is whether one, in a high-dimensional simulator-based setting, can construct practical inferential tools with finite-sample guarantees of frequentist coverage. Frequentist procedures have undoubtedly played an important role in many fields: In high energy physics (HEP), for instance, classical statistical techniques (e.g., hypothesis testing for signal detection) have resulted in discoveries of new physics and other successful applications (Feldman and Cousins, 1998; Cranmer, 2015; Tanabashi et al., 2018; Cousins, 2018). Even though controlling type I error probabilities is important in these applications, most simulator-based methods do not have theoretical guarantees on validity or power beyond low-dimensional data settings and large sample theory assumptions

(Feldman and Cousins, 1998). For example, some recent LFI methods (Frate et al., 2017; Brehmer et al., 2018, 2020b) are able to estimate likelihood functions for high-dimensional data, but these methods either assume that the log-likelihood ratio has an asymptotic chi-squared distribution, or rely on Monte Carlo samples at fixed parameter settings for inference (an approach that does not scale to computations of frequentist confidence sets). Ideally, a unified LFI approach should

- be *computationally efficient* in terms of the number of required simulations,
- handle *high-dimensional data* from different sources (without, e.g., predefined summary statistics),
- produce hypothesis tests and confidence sets that are *valid*; that is, have the nominal type I error or confidence level,
- produce hypothesis tests with *high power* or, equivalently, confidence sets with a small expected size,
- provide *diagnostics* for checking empirical coverage or for checking how well the estimated likelihood fits simulated data.

In this chapter, we present a statistical framework for LFI that unifies classical statistics with modern machine learning (e.g., deep generative models, neural network classifiers, and quantile regression) to achieve the following goals:

(i) provide confidence sets and hypothesis tests with finite-sample guarantees of frequentist coverage (nominal type I error) and power,

(ii) provide rigorous diagnostics for checking empirical coverage over the entire parameter space, and

(iii) present practical, easily extensible procedures that scale with both feature and parameter dimension.

At the heart of our proposed framework is the Neyman construction of confidence sets (Figure 2.2) — a procedure that is widely known and cited, but which nevertheless has not translated to practical algorithms for complex data settings beyond asymptotic approximations. The main bottleneck has been that the construction of confidence sets requires one to consider the null hypothesis $H_0 : \theta = \theta_0$ versus the alternative hypothesis $H_0 : \theta \neq \theta_0$ for every θ_0-value in the entire parameter space: The critical value (for Type I control) and the p-value will depend on θ_0. On a similar note, scientists have long recognized the importance of checking coverage of constructed confidence sets over all parameters but computationally this has been a real challenge; see, e.g., Cousins 2018, Section 13, for a discussion of evaluation of coverage with toy MC simulation.

This chapter introduces inferential and diagnostic tools that scale with both parameter and feature dimension. Our main observation is that key quantities of interest in frequentist statistical inference — test statistics, critical values, p-values and confidence set coverage — are conditional functions of the (unknown) parameter, and generally vary smoothly over the parameter space. As a result, one can leverage machine learning methods and data simulated in the neighborhood of a parameter to improve estimates of quantities of interest with fewer total simulations: We propose probabilistic classification for computing test statistics, quantile regression for estimating critical values, and regression for computing p-values and for checking empirical coverage.

Figure 2.1 shows our inference machinery. It has three main components, schematically represented as three branches in the diagram: First, we simulate a sample $\mathcal{T}$ to train a probabilistic classifier for learning a parametrized "odds function" (middle branch); as we shall see, this function can be used to construct meaningful test statistics (Section 2.2). We then simulate a second sample $\mathcal{T}^{'}$ to estimate critical values or p-values (left branch) for efficient construction of confidence sets (Section 2.3). Finally, we create a third sample $\mathcal{T}^{''}$ to assess empirical coverage across the parameter space (right branch; Section 2.4). Each component of our inference machinery is modular. Of particular note, is that (i) any test statistic defined in an LFI setting can be used to construct confidence sets with finite validity via the left branch, and (ii) our diagnostic procedures can provide insights to any LFI inference method via the right branch.

In this chapter we provide two novel odds-based test statistics: the `ACORE` and `BFF` test statistics. Both statistics are based on a parametrized odds function, but whereas `ACORE` eliminates the parameter θ by maximization, `BFF` averages over composite hypotheses.

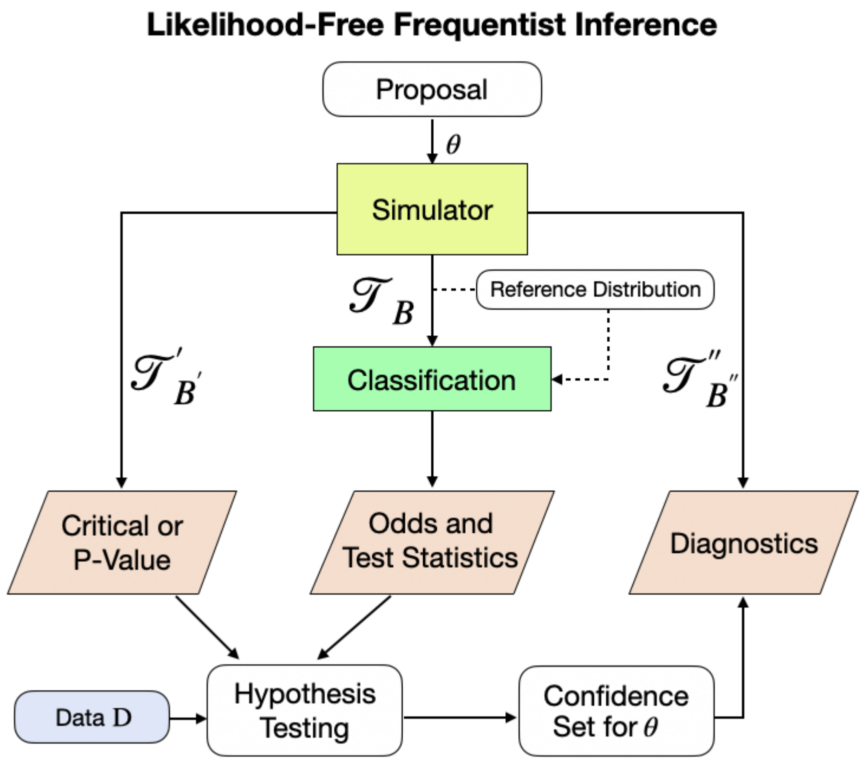


Figure 2.1: Schematic diagram of likelihood-free frequentist inference. The simulator provides synthetic observable data $\mathcal{T}_B$ for learning a parametrized odds function via probabilistic classification. The simulator also generates a separate sample $\mathcal{T}'_{B'}$ for learning critical values or p-values as a function of $\theta \in \Theta$. Once data D^{obs} are observed, the `BFF` or `ACORE` statistics can be used to construct hypothesis tests and confidence sets for θ. Our framework also provides diagnostics for computing the empirical coverage of constructed confidence sets as a function of the (unknown) parameter θ. The three main parts of the inference machinery (critical or p-value estimation, odds estimation, diagnostics) are separate modules. Each module leverages machine learning methods in the training phase and is amortized, i.e., they perform inference on new data without having to be retrained.

When the odds function is well-estimated for every θ and $\mathbf{x}$, the `ACORE` statistic is equal to the likelihood ratio statistics, and `BFF` is equal to the Bayes factor (Jeffreys, 1935, 1961). Although Bayes is included in the name of our test statistic, we here advocate for the use Bayes factor as a frequentist test statistic in likelihood-free inference.

2.1 Statistical Inference in a Traditional Setting

We begin by reviewing elements of traditional statistical inference that play a key role in our framework for likelihood-free frequentist inference.

Equivalence of tests and confidence sets. A classical approach to constructing a confidence set for an unknown parameter $\theta \in \Theta$ is to invert a series of hypothesis tests (Neyman, 1937): Suppose that for each possible value $\theta_0 \in \Theta$, there is a level α test δ_{θ_0} of

$$H_{0,\theta_0} : \theta = \theta_0 \quad \text{versus} \quad H_{1,\theta_0} : \theta \neq \theta_0; \tag{2.1}$$

that is, a test δ_{θ_0} where the type I error (the probability of erroneously rejecting a true null hypothesis H_{0,θ_0}) is no larger than α. For observed data $\mathcal{D} = D$, now define $R(D)$ as the set of all parameter values $\theta_0 \in \Theta$ for which the test δ_{θ_0} does not reject H_{0,θ_0}. Then, by construction, the random set $R(\mathcal{D})$ satisfies

$$\mathbb{P}\left[\theta_0 \in R(\mathcal{D}) \mid \theta = \theta_0\right] \geq 1 - \alpha$$

for all $\theta_0 \in \Theta$. That is, $R(\mathcal{D})$ defines a $(1-\alpha)$ *confidence set* for θ. Similarly, we can define a test with a desired significance level from a confidence set with a certain coverage.

Likelihood ratio test. A general form of hypothesis tests that often leads to high power is the likelihood ratio test (LRT). Consider testing

$$H_0 : \theta \in \Theta_0 \quad \text{versus} \quad H_1 : \theta \in \Theta_1, \tag{2.2}$$

where $\Theta_1 = \Theta \setminus \Theta_0$. For the *likelihood ratio (LR) statistic*,

$$\text{LR}(\mathcal{D}; \Theta_0) = \log \frac{\sup_{\theta \in \Theta_0} \mathcal{L}(\mathcal{D}; \theta)}{\sup_{\theta \in \Theta} \mathcal{L}(\mathcal{D}; \theta)}, \tag{2.3}$$

the LRT of hypotheses (2.2) rejects H_0 when $\text{LR}(D; \Theta_0) < C$ for some constant C.

Figure 2.2 illustrates the construction of confidence sets for θ from level α likelihood ratio tests (2.1). The critical value for each such test δ_{θ_0} is $C_{\theta_0} = \sup \{C : \mathbb{P}(\mathrm{LR}(\mathcal{D};\theta_0) < C \mid \theta = \theta_0) \leq \alpha\}$.

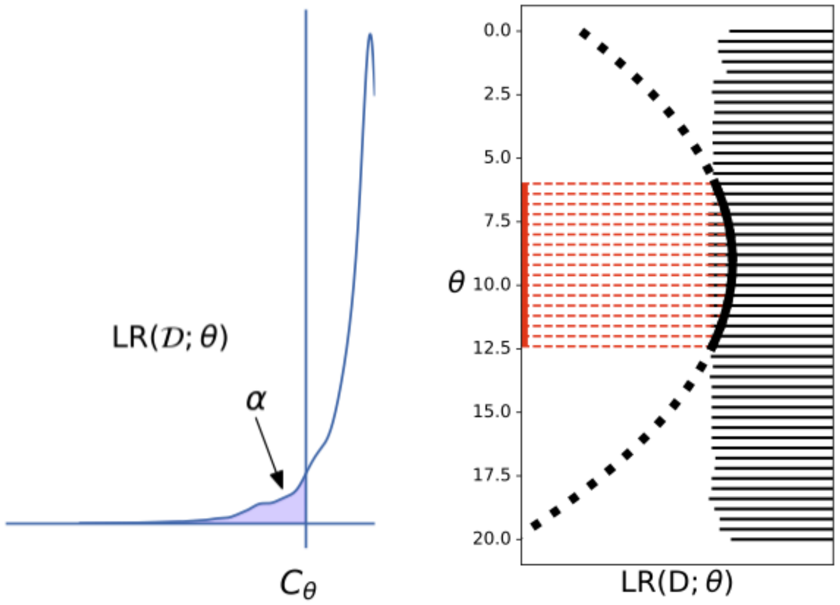


Figure 2.2: Constructing confidence intervals from hypothesis tests. *Left:* For each $\theta \in \Theta$, we find the critical value C_θ that rejects the null hypothesis $H_{0,\theta}$ at level α; that is, C_θ is the α-quantile of the distribution of the likelihood ratio statistic $\mathrm{LR}(\mathcal{D};\theta)$ under the null. *Right:* The horizontal lines represent the acceptance region for each $\theta \in \Theta$. Suppose we observe data $\mathcal{D} = D$. The confidence set for θ (indicated with the red line) consists of all θ-values for which the observed test statistic $\mathrm{LR}(D;\theta)$ (indicated with the black curve) falls in the acceptance region.

Bayes factor. Let π be a probability measure over the parameter space Θ. The Bayes factor (Jeffreys, 1935, 1961) for comparing the hypothesis $H_0 : \theta \in \Theta_0$ to its complement, the alternative H_1, is a likelihood ratio of the marginal likelihood of the two hypotheses:

$$\mathrm{BF}(\mathcal{D};\Theta_0) \equiv \frac{\mathbb{P}(\mathcal{D}|H_0)}{\mathbb{P}(\mathcal{D}|H_1)} = \frac{\int_{\Theta_0} \mathcal{L}(\mathcal{D};\theta) d\pi_0(\theta)}{\int_{\Theta_1} \mathcal{L}(\mathcal{D};\theta) d\pi_1(\theta)}, \tag{2.4}$$

where π_0 is the restriction of π_1 to Θ_0.

The Bayes factor is often used as a Bayesian alternative to significance testing as it quantifies the change in the odds in favor of H_0 when going from the prior to the posterior: $\frac{\mathbb{P}(H_0|\mathcal{D})}{\mathbb{P}(H_1|\mathcal{D})} = \mathrm{BF}(\mathcal{D};\Theta_0)\frac{\mathbb{P}(H_0)}{\mathbb{P}(H_1)}$.

2.2 Test Statistics Based on Parameterized Odds

In the typical LFI setting, we cannot directly evaluate the likelihood ratio $\mathrm{LR}(\mathcal{D};\Theta_0)$, or even the likelihood $\mathcal{L}(\mathcal{D};\theta)$. A simulator-based approach (`ACORE` or `BFF`; Figure 2.1) can nevertheless lead to hypothesis tests and confidence sets with good frequentist properties. The assumptions are access to (i) a "high-fidelity" forward simulator, also denoted by F_θ, that can simulate observable data, (ii) a reference distribution G with larger support than F_θ for all $\theta \in \Theta$, and (iii) a probabilistic classifier that discriminates samples from F_θ and G.

2.2.1 Parametrized Odds for Labeled Samples

We start by generating a labeled sample $\mathcal{T}_B = \{\theta_i, \mathbf{X}_i, Y_i\}_{i=1}^B$ to compare data from the simulator F_θ with data from the reference distribution G. Here, $\theta \sim \pi_\Theta$ (a fixed proposal distribution over Θ), the "label" $Y \sim \mathrm{Ber}(p)$, $\mathbf{X}|\theta, Y = 1 \sim F_\theta$, and $\mathbf{X}|\theta, Y = 0 \sim G$. We then define the odds at θ and fixed $\mathbf{x}$ as

$$\mathbb{O}(\mathbf{x};\theta) := \frac{\mathbb{P}(Y = 1|\theta, \mathbf{x})}{\mathbb{P}(Y = 0|\theta, \mathbf{x})}.$$

One way of interpreting the odds $\mathbb{O}(\theta, \mathbf{X})$ is to regard it as a measure of the chance that $\mathbf{X}$ was generated from F_θ rather than from G. That is, a large odds $\mathbb{O}(\theta, \mathbf{X})$ reflects the fact that it is plausible that $\mathbf{X}$ was generated from F_θ (instead of G). We call G a "reference distribution" as we are comparing F_θ for different θ with this distribution.

The odds $\mathbb{O}(\theta, \mathbf{x})$ with $\theta \in \Theta$ as a parameter can be learnt with a probabilistic classifier, such as a neural network with a softmax layer, suitable for the data $\mathbf{X}$ at hand. Algorithm 1 provides details on how to create the training sample $\mathcal{T}_B$ for estimating odds. Out of the total training sample size B, a proportion p is generated by the stochastic forward simulator F_θ at different parameter values θ, while the remainder is sampled from a reference distribution G. Note that G can be any distribution that dominates F_θ. If G is the marginal distribution $F_\mathbf{x}$, there is the nice property that the denominator of the `BFF` statistic is 1, allowing us to bypass calculating the denominator term. For all experiments in this paper, we use p=1/2 and $G = F_\mathbf{X}$, where $F_\mathbf{X}$ is the marginal of F_θ with respect to π_Θ.

Algorithm 1 Generate a labeled sample of size B for estimating odds

Input: stochastic forward simulator F_θ; reference distribution G; proposal distribution π_Θ over parameter space; training sample size B; parameter p of Bernoulli distribution

Output: labeled training sample

```
 1: Set $\mathcal{T} \leftarrow \emptyset$
 2: for $i$ in $\{1, ..., B\}$ do
 3:   Draw parameter value $\theta_i \sim \pi_\Theta$
 4:   Draw $Y_i \sim \text{Ber}(p)$
 5:   if $Y_i == 1$ then
 6:     Draw sample $\mathbf{X}_i \sim F_{\theta_i}$
 7:   else
 8:     Draw sample $\mathbf{X}_i \sim G$
 9:   end if
10:   $\mathcal{T} \leftarrow \mathcal{T} \cup (\theta_i, \mathbf{X}_i, Y_i)$
11: end for
12: return $\mathcal{T} = \{\theta_i, \mathbf{X}_i, Y_i\}_{i=1}^B$
```

Algorithm 2 Sample from marginal distribution $F_{\mathbf{X}}$

Input: stochastic forward simulator F_θ; proposal distribution π_Θ over parameter space

Output: sample $\mathbf{X}_i$ from marginal distribution

```
1: Draw parameter value $\theta_i \sim \pi_\Theta$
2: Draw sample $\mathbf{X}_i \sim F_{\theta_i}$
3: return $\mathbf{X}_i$
```

For testing $H_{0,\theta_0} : \theta = \theta_0$ versus all alternatives $H_{1,\theta_0} : \theta \neq \theta_0$, we consider two test statistics: `ACORE` and `BFF`. Both statistics are based on $\mathbb{O}(\mathbf{X};\theta)$, but whereas `ACORE` eliminates the parameter θ by maximization, `BFF` averages over composite hypotheses.

2.2.2 `ACORE` by Maximization

The `ACORE` statistic for testing (2.1) is given by

$$\begin{aligned}\Lambda(\mathcal{D};\theta_0) &:= \log \frac{\prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};\theta_0)}{\sup_{\theta\in\Theta}\prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};\theta)} \\ &= \inf_{\theta_1\in\Theta}\sum_{i=1}^n \log\left(\mathbb{OR}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right), \qquad (2.5)\end{aligned}$$

where the odds ratio

$$\mathbb{OR}(\mathbf{x};\theta_0,\theta_1) := \frac{\mathbb{O}(\theta_0;\mathbf{x})}{\mathbb{O}(\theta_1;\mathbf{x})}$$

at $\theta_0, \theta_1 \in \Theta$ measures the plausibility that $\mathbf{x}$ was generated from θ_0 rather than θ_1.

We use $\widehat{\Lambda}_B(\mathcal{D}; \theta_0)$ to denote the `ACORE` statistic based on $\mathcal{T}_B$ and estimated odds $\widehat{\mathbb{O}}(\theta_0; \mathbf{x})$. When $\widehat{\mathbb{O}}(\theta_0; \mathbf{x})$ is well-estimated for every θ and $\mathbf{x}$, the `ACORE` statistic $\widehat{\Lambda}_B(\mathcal{D}; \theta_0)$ is the same as the likelihood ratio statistic $\mathrm{LR}(\mathcal{D}; \theta_0)$:

Proposition 1 (Fisher Consistency)**.** *If* $\widehat{\mathbb{P}}(Y = 1|\theta, \mathbf{x}) = \mathbb{P}(Y = 1|\theta, \mathbf{x})$ *for every* θ *and* $\mathbf{x}$*, then the* `ACORE` *test statistic (2.5) is the likelihood ratio statistic (Equation 2.3).*

Proof of Proposition 1. If $\widehat{\mathbb{P}}(Y = 1|\theta, \mathbf{x}) = \mathbb{P}(Y = 1|\theta, \mathbf{x})$, then $\widehat{\mathbb{OR}}(\mathbf{x}; \theta_0, \theta_1) = \mathbb{OR}(\mathbf{x}; \theta_0, \theta_1)$. By Bayes rule and construction (Algorithm 1),

$$\mathbb{O}(\mathbf{x}; \theta) := \frac{\mathbb{P}(Y = 1|\theta, \mathbf{x})}{\mathbb{P}(Y = 0|\theta, \mathbf{x})} = \frac{f(\mathbf{x}|\theta)p}{g(\mathbf{x})(1-p)}.$$

Thus, the odds ratio at $\theta_0, \theta_1 \in \Theta$ is given by

$$\mathbb{OR}(\mathbf{x}; \theta_0, \theta_1) = \frac{f(\mathbf{x}|\theta_0)}{f(\mathbf{x}|\theta_1)},$$

and therefore

$$\begin{aligned}
\tau(D; \Theta_0) &= \sup_{\theta_0 \in \Theta_0} \inf_{\theta_1 \in \Theta} \sum_{i=1}^{n} \left(\log \widehat{\mathbb{OR}}(\mathbf{x}_i^{\text{obs}}; \theta_0, \theta_1) \right) \\
&= \sup_{\theta_0 \in \Theta_0} \inf_{\theta_1 \in \Theta} \sum_{i=1}^{n} \log \frac{f(\mathbf{x}_i^{\text{obs}}|\theta_0)}{f(\mathbf{x}_i^{\text{obs}}|\theta_1)} \\
&= \sup_{\theta_0 \in \Theta_0} \inf_{\theta_1 \in \Theta} \log \left(\frac{\mathcal{L}(D; \theta_0)}{\mathcal{L}(D; \theta_1)} \right) \\
&= \Lambda(D; \Theta_0).
\end{aligned}$$

□

2.2.3 `BFF` by Averaging

Because the `ACORE` statistics in Equation 2.5 involves taking the supremum (or infimum) over Θ, it may not be practical in high dimensions. Hence, in this work, we propose an alternative statistic for testing (2.1) based on averaged odds:

$$\tau(\mathcal{D};\theta_0) := \frac{\prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};\theta_0)}{\int_\Theta \left(\prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};\theta)\right) d\pi(\theta)}. \tag{2.6}$$

Let $\widehat{\tau}_B(\mathcal{D};\theta_0)$, or $\widehat{\tau}$ for short, denote estimates based on $\mathcal{T}_B$ and $\widehat{\mathbb{O}}(\theta_0;\mathbf{x})$. If the probabilities learned by the classifier are well estimated, then the averaged odds test statistic is exactly the Bayes Factor:

Proposition 2 (Fisher Consistency)**.**

If $\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{x}) = \mathbb{P}(Y=1|\theta,\mathbf{x})$ *for every* θ *and* $\mathbf{x}$*, then* $\widehat{\tau}(\mathcal{D};\theta_0)$ *is the Bayes Factor* $BF(\mathcal{D};\theta_0)$*.*

Proof of Proposition 2. By Bayes rule,

$$\mathbb{O}(\mathbf{x};\theta) := \frac{\mathbb{P}(Y=1|\theta,\mathbf{x})}{\mathbb{P}(Y=0|\theta,\mathbf{x})} = \frac{f(\mathbf{x}|\theta)p}{g(\mathbf{x})(1-p)}.$$

If $\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{x}) = \mathbb{P}(Y=1|\theta,\mathbf{x})$, then $\widehat{\mathbb{O}}(\mathbf{x};\theta_0) = \mathbb{O}(\mathbf{x};\theta_0)$. Therefore,

$$\begin{aligned}
\widehat{\tau}(\mathcal{D};\theta_0) :=& \frac{\prod_{i=1}^n \widehat{\mathbb{O}}(\mathbf{X}_i^{\text{obs}};\theta_0)}{\int_\Theta \left(\prod_{i=1}^n \widehat{\mathbb{O}}(\mathbf{X}_i^{\text{obs}};\theta)\right) d\pi(\theta)} \\
=& \frac{\prod_{i=1}^n \frac{f(\mathbf{X}_i^{\text{obs}}|\theta_0)}{g(\mathbf{X}_i^{\text{obs}})}}{\int_\Theta \left(\prod_{i=1}^n \frac{f(\mathbf{X}_i^{\text{obs}}|\theta)}{g(\mathbf{X}_i^{\text{obs}})}\right) d\pi(\theta)} \\
=& \frac{\prod_{i=1}^n f(\mathbf{X}_i^{\text{obs}}|\theta_0)}{\int_\Theta \left(\prod_{i=1}^n f(\mathbf{X}_i^{\text{obs}}|\theta)\right) d\pi(\theta)} \\
=& \frac{\mathcal{L}(\mathcal{D};\theta_0)}{\int_\Theta \mathcal{L}(\mathcal{D};\theta)\pi(\theta)d\theta} \\
=& \text{BF}(\mathcal{D};\theta_0).
\end{aligned}$$

□

As a reminder, we are using the Bayes Factor as a frequentist test statistic. Hence, our term *Bayes Factor Frequentist (BFF)* statistic for τ and $\widehat{\tau}$.

2.3 Efficient Construction of Finite-Sample Confidence Sets

Although the Neyman construction of confidence sets (Figure 2.2) is widely known, this procedure has not translated to practical machine learning algorithms for complex data settings. Here we address the question: "How do we *efficiently* estimate critical values and significance probabilities (p-values) of a test when we do not know the distribution of our test statistic, and cannot rely on large-sample theory approximations?"

Simulation-based approaches are often used to compute rejection probabilities and critical values in lieu of large-sample theory approximations. Typically, such simulations compute a separate Monte Carlo simulation at each fixed $\theta_0 \in \Theta$ on, e.g., a fine enough grid in parameter space. That is, the convention is to rely solely on sample points generated at fixed θ_0 to estimate the rejection probabilities. What we do instead is to treat the critical value C or the p-value as functions that vary smoothly with the parameter θ; we then estimate the parametrized functions C_{θ_0} and $p(D; \theta_0)$ for any $\theta_0 \in \Theta$ via quantile regression (Section 2.3.1) and regression (Section 2.3.2), respectively. We summarise the procedure of constructing a confidence interval in Algorithm 3.

2.3.1 The Critical Value via Quantile Regression

Suppose that we reject the null hypothesis if the `BFF` test statistic (or `ACORE` , equivalently) is smaller than some constant C. To achieve a test with a desired level of significance α, we need (for maximum power) the largest C that satisfies

$$\mathbb{P}\left(\tau(\mathcal{D}; \theta_0) < C \mid \theta\right) \leq \alpha. \tag{2.7}$$

However, we cannot explicitly compute the critical value C or the rejection probability as we do not know the distribution of the test statistic τ.

Simulation-based approaches are often used to compute rejection probabilities and critical values in lieu of large-sample theory approximations. Typically, such simulations compute a separate Monte Carlo simulation at each fixed $\theta \in \Theta_0$ on, e.g., a fine enough grid on θ. That is, the convention is to rely solely on sample points generated at fixed θ to estimate the rejection probabilities $\mathbb{P}(\tau(\mathcal{D}; \theta_0) < C|\theta)$. Here we propose to estimate the

Algorithm 3 Construct confidence set for θ with coefficient $\gamma = 1 - \alpha$

Input: stochastic forward simulator F_θ; proposal distribution π over Θ; parameter p of Bernoulli distribution; sample size B (for estimating odds ratios); sample size B' (for estimating critical values or p-values); probabilistic classifier; observed data $D = \left\{\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}\right\}$; desired level $\alpha \in (0,1)$; number of parameter values at which to evaluate confidence set n_{grid}; confidence set estimation strategy `eval` (either `p-values` or `crit-values`)

Output: θ-values in confidence set

```
 1: // Estimate odds
 2: Generate labeled sample $\mathcal{T}_B$ according to Algorithm 1
 3: Apply probabilistic classifier to $\mathcal{T}_B$ to learn class posterior probabilities, $\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X})$,
    for all $\theta = (\theta, \psi) \in \Theta$ and $\mathbf{X} \in \mathcal{X}$
 4: Let the estimated odds $\widehat{\mathbb{O}}(\mathbf{X};\theta) = \frac{\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X})}{\widehat{\mathbb{P}}(Y=0|\theta,\mathbf{X})}$
 5: // Test statistic $\tau$
 6: Define $\tau(\mathcal{D};\theta) \leftarrow \tau(\mathcal{D};\theta)$ for every $\theta$
 7: // Find parameter set for which the test $\delta_{\theta_0}$ does not reject $\theta = \theta_0$
 8: $\mathrm{L}_\Theta \leftarrow$ lattice over $\Theta$ with $n_{\text{grid}}$ elements
 9: Set $S \leftarrow \emptyset$
10: if eval == crit-values then
11:     Estimate critical value $\widehat{C}_\theta \leftarrow \widehat{F}^{-1}_{\tau|\theta}(\alpha \mid \theta)$ for every $\theta$          (Algorithm 4)
12: else if eval == p-values then
13:     Estimate p-value $\widehat{p}(D;\theta) \leftarrow \widehat{\mathbb{E}}\left[\mathbb{I}\left(\widehat{\tau}(\mathcal{D};\theta) < \tau(D;\theta)\right) \mid \theta\right]$ for every $\theta$    (Algorithm 5)
14: end if
15: for $\theta_0 \in \mathrm{L}_\Theta$ do
16:     Compute the observed profiled test statistic $\widehat{\tau}^{\text{obs}} \leftarrow \widehat{\tau}(D;\theta_0)$
17:     if eval == crit-values then
18:         if $\widehat{\tau}^{\text{obs}} > \widehat{C}_{\theta_0}$ then
19:             $S \leftarrow S \cup \{\theta_0\}$
20:         end if
21:     else if eval == p-values then
22:         if $\widehat{p}(D;\theta_0) > \alpha$ then
23:             $S \leftarrow S \cup \{\theta_0\}$
24:         end if
25:     end if
26: end for
27: return
```

critical values C for all $\theta \in \Theta_0$ and significance levels $\alpha \in [0,1]$ simultaneously. At the heart of our approach is the key observation that the rejection probability $\mathbb{P}(\tau(\mathcal{D};\Theta_0) < C|\theta)$ is a conditional cumulative distribution function, which in many settings varies smoothly as a function of θ and C. Thus, similar to how we estimate odds for the `ACORE` and `BFF` statistic, one can use data generated in the neighborhood of θ to improve estimates of our quantities of interest at any θ. This is what a quantile regression implicitly does to estimate C (e.g., Meinshausen 2006; Koenker et al. 2017).

Algorithm 4 outlines the details of the procedure for estimating C. In brief, we use a training sample $\mathcal{T}'_{B'} = \{(\theta_i, \tau_i)\}_{i=1}^{B'}$ (independent of $\mathcal{T}_B$) to estimate the α-conditional quantile $c_\alpha(\theta)$ defined by $\mathbb{P}(\tau \leq c_\alpha(\theta) \mid \theta) = \alpha$. Let $\widehat{c}_\alpha(\theta)$ be the estimate of $c_\alpha(\theta)$ from a quantile regression of τ on θ. By (2.7), our estimate of the critical value C is $\widehat{C} = \inf_{\theta \in \Theta_0} \widehat{c}_\alpha(\theta)$. As we shall see, even if the odds are not well estimated, tests and confidence regions based on estimated odds are still valid as long as the thresholds are well estimated: Theorem 2.1 shows that the sample size $B^{'}$ controls the type I error of the test, regardless of the observed data sample size n.

To test a composite null hypothesis $H_0 : \theta \in \Theta_0$ versus $H_1 : \theta \in \Theta_1$, we use the cutoff $\widehat{C}_{\Theta_0} := \inf_{\theta \in \Theta_0} \widehat{C}_\theta$. This cutoff leads to a valid test (control of type I error as $B' \to \infty$) regardless of whether the test statistic τ is well estimated or not, and regardless of the data sample size n.

Algorithm 4 Estimate the critical values C_{θ_0} for a level-α test of $H_{0,\theta_0} : \theta = \theta_0$ vs. $H_{1,\theta_0} : \theta \neq \theta_0$ for all $\theta_0 \in \Theta$ simultaneously

Input: stochastic forward simulator F_θ; sample size B' for training quantile regression estimator; π (a fixed proposal distribution over the full parameter space Θ); test statistic $\widehat{\tau}$; quantile regression estimator; desired level $\alpha \in (0, 1)$

Output: estimated critical values $\widehat{C}_\theta$ for all $\theta = \theta_0 \in \Theta$

```
1: Set $\mathcal{T}' \leftarrow \emptyset$
2: for i in {1,...,B'} do
3:    Draw parameter $\theta_i \sim \pi$
4:    Draw sample $\mathbf{X}_{i,1}, \ldots, \mathbf{X}_{i,n} \overset{iid}{\sim} F_{\theta_i}$
5:    Compute test statistic $\widehat{\tau}_i \leftarrow \widehat{\tau}((\mathbf{X}_{i,1}, \ldots, \mathbf{X}_{i,n}); \theta_i)$
6:    $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{(\theta_i, \widehat{\tau}_i)\}$
7: end for
8: Use $\mathcal{T}'$ to learn parametrized function $\widehat{C}_\theta := \widehat{F}^{-1}_{\widehat{\tau}|\theta}(\alpha|\theta)$ via quantile regression of $\widehat{\tau}$ on $\theta$
   return $\widehat{C}_{\theta_0}$
```

2.3.2 The P-Value via Regression

If the data D are observed beforehand, we can alternatively compute p-values for each hypothesis $H_{0,\theta_0} : \theta = \theta_0$, that is,

$$p(D;\theta_0) := \mathbb{P}_{\mathcal{D}|\theta_0,\mathcal{T}_B}(\widehat{\tau}(\mathcal{D};\theta_0) < \widehat{\tau}(D;\theta_0)),$$

where $p(D;\theta_0)$ can be used to test hypothesis and create confidence sets for any desired α. These p-values can be estimated by noticing that they are a regression $\mathbb{E}[Z|\theta_0]$ of the random variable $Z := \mathbb{I}\left(\widehat{\tau}(\mathcal{D};\theta_0) < \widehat{\tau}(D;\theta_0)\right)$ on θ_0. Thus, we can as in Algorithm 5 generate a train sample $\mathcal{T}' = \{(Z_1,\theta_1),\ldots,(Z_{B'},\theta_{B'})\}$ and then estimate p-values for all $\theta_0 \in \Theta$ simultaneously. An advantage of using a regression instead of a quantile regression approach (as in Section 2.3.1) is that we can take advantage of the many existing regression methods.

For testing the composite null hypothesis $H_0 : \theta \in \Theta_0$ versus $H_1 : \theta \in \Theta_1$, we use

$$\widehat{p}(D;\Theta_0) := \sup_{\theta\in\Theta_0} \widehat{p}(D;\theta)$$

with

$$\widehat{p}(D;\theta) := \widehat{\mathbb{P}}_{\mathcal{D}|T,\theta_0}(\widehat{\tau}(\mathcal{D};\theta_0) < \widehat{\tau}(D;\theta_0)), \tag{2.8}$$

where the odds statistic $\widehat{\tau}$ is computed given train sample $\mathcal{T}_B = T$.

2.4 Confidence Sets and Diagnostics

After the parametrized `ACORE` statistic and the critical values have been estimated, it is important to check whether the resulting confidence sets indeed are valid or, equivalently, if the resulting hypothesis tests have the nominal significance level. We also want to identify regions in parameter space where we clearly overcover. That is, the two main questions are: (i) do the constructed confidence sets satisfy

$$\mathbb{P}\left[\theta_0 \in R(\mathcal{D}) \mid \theta = \theta_0\right] \geq 1-\alpha,$$

Algorithm 5 Estimate the p-values $p(D;\theta_0)$, given observed data D, for a level-α test of $H_{0,\theta_0}: \theta = \theta_0$ vs. $H_{1,\theta_0}: \theta \neq \theta_0$ for all $\theta_0 \in \Theta$ simultaneously.

Input: observed data D; stochastic forward simulator F_θ; sample size B' for p-value estimation; π (a fixed proposal distribution over the full parameter space Θ); test statistic $\widehat{\tau}$; regression estimator m

Output: estimated p-value $\widehat{p}(D;\theta)$ for all $\theta = \theta_0 \in \Theta$

```
1: Set $\mathcal{T}' \leftarrow \emptyset$
2: for i in {1,...,B'} do
3:     Draw parameter $\theta_i \sim \pi$
4:     Draw sample $\mathbf{X}_{i,1}, \ldots, \mathbf{X}_{i,n} \overset{iid}{\sim} F_{\theta_i}$
5:     Compute test statistic $\widehat{\tau}_i \leftarrow \widehat{\tau}((\mathbf{X}_{i,1}, \ldots, \mathbf{X}_{i,n}); \theta_i)$
6:     Compute indicator $Z_i \leftarrow \mathbb{I}\left(\widehat{\tau}_i < \widehat{\tau}(D;\theta_i)\right)$
7:     $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{(\theta_i, Z_i)\}$
8: end for
9: Use $\mathcal{T}'$ to learn parametrized function $\widehat{p}(D;\theta) := \widehat{\mathbb{E}}[Z|\theta]$ via regression of $Z$ on $\theta$ using
   regression estimator $m$
   return $\widehat{p}(D;\theta_0)$
```

or alternatively,

$$\widehat{R}(D) = \{\theta_0 \in \Theta \; | \widehat{p}(D;\theta_0) > \alpha\}$$

for every $\theta_0 \in \Theta$, and (ii) how close is the actual coverage to the nominal confidence level $1-\alpha$?

To answer these questions, we propose a goodness-of-fit procedure where we draw B'' new samples from the simulator given θ, construct a confidence set for each sample, and then check which computed regions include the "true" θ. More specifically: we generate a set $\mathcal{T}''_{B''} = \{(\theta'_1, \mathcal{D}'_1), \ldots, (\theta'_{B''}, \mathcal{D}'_{B''})\}$, where $\theta'_i \sim r_\Theta$ and $\mathcal{D}'_i$ is a sample of size n of i.i.d. observable data from $F_{\theta'_i}$. We then define

$$W_i := \mathbb{I}\left(\theta'_i \in R(\mathcal{D}'_i)\right),$$

where $R(\mathcal{D}'_i)$ is the confidence set for θ for data $\mathcal{D}'_i$. If R has the correct coverage, then

$$\mathbb{P}(W_i = 1|\theta_i) \geq 1 - \alpha.$$

We can estimate the probability $\mathbb{P}(W_i = 1|\theta_i)$ using any probabilistic classifier; some methods also provide confidence bands that assess the uncertainty in estimating this quantity (Eubank and Speckman, 1993; Claeskens et al., 2003; Krivobokova et al., 2010). By comparing the estimated probability to $1-\alpha$, we have a diagnostic tool for checking how close we are to the nominal confidence level over the entire parameter space Θ. See Figure 2.3 for an example. Note that our procedure parametrizes the coverage of the confidence set as a function of the true parameter value. This is in contrast to other goodness-of-fit techniques (e.g., Cook et al. 2006; Bordoloi et al. 2010; Talts et al. 2018; Schmidt et al. 2020) that only check for *marginal* coverage, i.e., $n^{-1}\sum_{i=1}^{n} W_i \geq 1-\alpha$. In addition, as shown in Theorem 2.1 the random set $\widehat{R}(\mathcal{D})$ has the nominal $1-\alpha$ coverage as $B' \to \infty$ regardless of the data sample size n.

Finally, note that our diagnostic procedure relies on assuming that the confidence set varies smoothly across the parameter space. This assumption might break down in cases when the observable data are discrete, especially for small values of the true parameters. The estimated coverage would then be an interpolated version of the true coverage across the parameter space. Heinrich (2003, Section 4) show that likelihood-ratio based confidence sets undercover the true parameter value for Poisson data when the true parameter is very small ($\theta < 2$), and in general does not vary continuously across the parameter space. This issue extends to other discrete distribution such as the binomial, hypergeometric and negative binomial distribution (see Casella and Berger 2002, Section 9.5.2); Blaker (2000) provides a solution to build valid confidence sets in these setting. For such reasons, one might prefer constructing confidence sets using the `BFF` rather than `ACORE` test statistic in the presence of low-count discrete data. Regardless of the test statistic used, our approach would still be able to highlight under-coverage issues provided a large enough B, $B^{'}$ and $B^{''}$; as an example, see Figure 2.11 (bottom panel) for small values of the signal parameter S (see Section 2.9.4 for more details). Alternatively, one could estimate coverage using a basis function that can capture the discrete nature of the true coverage function, such as Haar wavelets for instance (Haar, 1910).

2.5 Loss function

In our inference machinery, we use the cross-entropy loss to train probabilistic classifiers. In what follows we show the close connection between cross entropy loss and estimates odds.

Consider a sample point $\{\theta, \mathbf{x}, y\}$ generated according to Algorithm 1, and the estimated odds function $g(\theta, \mathbf{x}) := \widehat{\mathbb{P}}(Y=1|\theta,\mathbf{x})/\widehat{\mathbb{P}}(Y=0|\theta,\mathbf{x})$. Let p be a Ber(y) distribution, and q be a Ber $\left(\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{x})\right) = \text{Ber}\left(\frac{g(\theta,\mathbf{x})}{1+g(\theta,\mathbf{x})}\right)$ distribution. The *cross entropy* between p and q is given by

$$\begin{aligned} L_{\text{CE}}(g;\{\theta,\mathbf{x},y\}) &= -y\log\left(\frac{g(\theta,\mathbf{x})}{1+g(\theta,\mathbf{x})}\right) - (1-y)\log\left(\frac{1}{1+g(\theta,\mathbf{x})}\right) \\ &= -y\log\left(g(\theta,\mathbf{x})\right) + \log\left(1+g(\theta,\mathbf{x})\right). \end{aligned} \tag{2.9}$$

For every $\mathbf{x}$ and θ, the expected cross entropy $\mathbb{E}[L_{\text{CE}}(g;\{\theta,\mathbf{x},Y\})]$ is minimized by $g(\theta,\mathbf{x}) = \mathbb{O}(\theta,\mathbf{x})$. Thus we can measure the performance of an estimator g of the odds by the risk

$$R_{\text{CE}}(g) = \mathbb{E}[L_{\text{CE}}(g;\{\theta,\mathbf{X},Y\})].$$

The cross entropy loss is not the only loss function that is minimized by the true odds function, but it is usually easy to compute in practice. It is also well known that minimizing the cross entropy loss between the estimated distribution q and the true distribution p during training is equivalent to minimizing the Kullback-Leibler (KL) divergence between the two distributions, as

$$KL(p||q) = H(p,q) - H(p),$$

where $H(p,q)$ is the cross entropy and $H(p)$ is the entropy of the true distribution. By Gibbs' inequality (MacKay, 2002), we have that $KL(p||q) \geq 0$; hence the entropy $H(p)$ of the true distribution lower bounds the cross entropy with the minimum achieved when $p = q$. Hence, we can connect the cross entropy loss to the `ACORE` and `BFF` statistics.

Proposition 3. *If the probabilistic classifier achieves the minimum of the cross entropy loss, then the constructed `ACORE` statistic (2.5) is equal to the likelihood ratio statistic (2.3) and the constructed `BFF` statistic (2.6) is equal to the Bayes factor (2.4).*

Proof of Prop 3. The proof follows from Proposition 1 and 2, and the expected cross entropy loss is minimized if and only if $\widehat{\mathbb{O}}(\theta,\mathbf{x}) = \mathbb{O}(\theta,\mathbf{x})$. $\square$

In addition, we show that the convergence of the class posterior implies the convergence of the cross entropy to the entropy of the true distribution. This supports our decision to use the cross entropy loss when selecting the probabilistic classifier and sample size B.

Lemma 2.0.1. *If for every $\theta \in \Theta$*

$$q := \widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X}) \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} p := \mathbb{P}(Y=1|\theta,\mathbf{X}),$$

then $H(p,q) \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} H(p)$.

Proof of Lemma 2.0.1. We can rewrite the cross entropy $H(p,q)$ and entropy $H(p)$ as

$$\begin{aligned} H(p,q) &= -\sum_{y\in\{0,1\}} \int_{\mathcal{X}\times\Theta} p\log(q)\, \mathrm{d}\mathbb{P}(\mathbf{x},\theta), \\ H(p) &= -\sum_{y\in\{0,1\}} \int_{\mathcal{X}\times\Theta} p\log(p)\, \mathrm{d}\mathbb{P}(\mathbf{x},\theta). \end{aligned}$$

In addition, for any $(\mathbf{X},\theta)$, it also holds that $|q| \leq 1$. The lemma follows by combining the dominated convergence theorem with the continuous mapping theorem for the logarithm. □

In the special case where G is the marginal distribution of $F_\theta(\mathbf{x})$ with respect to π and we in addition assume that $\mathbf{x}$ contains all observations (that is, $\mathbf{X} = \mathcal{D}$), the denominator of the `BFF` statistic (2.6) is equal to one. The BFF test statistic then simply becomes the integrated odds (2.11). Hence, in addition to the standard cross-entropy loss (2.9), we propose an *integrated odds loss* function which is directly related to the BFF (integrated odds) statistic:

$$\mathcal{L}(\widehat{\mathbb{O}},\mathbb{O}) := \int \left(\widehat{\mathbb{O}}(\mathbf{x};\theta) - \mathbb{O}(\mathbf{x};\theta)\right)^2 dg(\mathbf{x})d\pi(\theta). \tag{2.10}$$

In Section 2.6, Theorem 3, we show that the power of the `BFF` test statistic is bounded by the integrated odds loss.

2.6 Theoretical Guarantees

In this section we provide theoretical guarantees on the inference machinery presented in this chapter. Section 2.6.1 shows that Algorithm 4 leads to valid hypothesis tests as long as B' is large enough (Theorem 2.1), and that the power of the test converges to the power of the LRT as B grows (Theorem 2.2). Section 2.6.2 proves the consistency of the p-value estimation method in Algorithm 5, while Section 2.6.3 provides theoretical guarantees for the power of `BFF`.

2.6.1 Critical Value Estimation

We denote convergence in probability and in distribution by $\xrightarrow{\mathbb{P}}$ and $\xrightarrow{\text{Dist}}$, respectively. We start by showing that our procedure leads to valid hypothesis tests (that is, tests that control the type I error probability) as long as B' in Algorithm 3 is large enough. In order to do so, we assume that the quantile regression estimator used in Algorithm 3 to estimate the critical values is consistent in the following sense:

Assumption 2.1. *Let $\widehat{F}_{B'}(\cdot|\theta)$ be the estimated cumulative distribution function of the test statistic τ conditional on θ based on a sample size B', and let $F(\cdot|\theta)$ be true conditional distribution. For every $\theta \in \Theta_0$, assume that the quantile regression estimator is such that*

$$\sup_{t\in\mathbb{R}} |\widehat{F}_{B'}(t|\theta) - F(t|\theta)| \xrightarrow[B' \longrightarrow \infty]{\mathbb{P}} 0$$

Under some conditions, Assumption 2.1 holds for instance for quantile regression forests (Meinshausen, 2006).

Next we show that, for every fixed training sample size B in Algorithm 1, Algorithm 3 yields a valid hypothesis test as $B' \to \infty$. The result holds even if the likelihood ratio statistic is not well estimated.

Theorem 2.1. *Let $C_{B,B'} \in \mathbb{R}$ be the critical value of the test based on the statistic $\tau = \tau_B$ for a training sample size B with critical value chosen according to Algorithm 3 for a fixed $\alpha \in (0,1)$. Assume the quantile estimator satisfies Assumption 2.1, and further assume either:*

- $|\Theta| < \infty$,

- *$|\Theta|$ is a compact subset of $\mathbb{R}^d$, and the function $g_{B'}(\theta) = \sup_{t\in\mathbb{R}} |\widehat{F}_{B'}(t|\theta) - F(t|\theta)|$ is continuous in θ and strictly decreasing in B',*

then

$$C_{B,B'} \xrightarrow[B'\longrightarrow\infty]{\mathbb{P}} C^*_B,$$

where C^*_B is such that

$$\sup_{\theta\in\Theta_0} \mathbb{P}(\tau_B \leq C^*_B|\theta) = \alpha.$$

Finally we show that as long as the probabilistic classifier is consistent and the critical values are well estimated (which holds for large B' according to Theorem 2.1), the power of the `ACORE` test converges to the power of the LRT as B grows. Equivalent theorems also apply to the `BFF` test statistics.

Theorem 2.2. *Let $\widehat{\phi}_{B,C_B}(\mathcal{D})$ be the test based on the statistic $\tau = \tau_B$ for a labeled sample size B with critical value $C_B \in \mathbb{R}$.[*] Moreover, let $\phi_{C^*}(\mathcal{D})$ be the likelihood ratio test with critical value $C^* \in \mathbb{R}$.[†] If, for every $\theta \in \Theta$,*

$$\widehat{\mathbb{P}}(Y = 1|\theta, \mathbf{X}) \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} \mathbb{P}(Y = 1|\theta, \mathbf{X}),$$

where $|\Theta| < \infty$, and $\widehat{C}_B$ is such that $\widehat{C}_B \xrightarrow[B\longrightarrow\infty]{Dist} C^*$, then, for every $\theta \in \Theta$,

$$\mathbb{P}\left(\widehat{\phi}_{B,\widehat{C}_B}(\mathcal{D}) = 1|\theta\right) \xrightarrow[B\longrightarrow\infty]{} \mathbb{P}\left(\phi_{C^*}(\mathcal{D}) = 1|\theta\right).$$

2.6.2 P-value estimation

We start by showing that the p-value estimation method described in Section 2.3.2 is consistent. The results shown here apply to any test statistic. That is, these results are not restricted to `BFF`.

We assume consistency in the sup norm of the regression method used to estimate the p-values:

[*]That is, $\widehat{\phi}_{B,C_B}(\mathcal{D}) = 1 \iff \tau_B(\mathcal{D};\Theta_0) < C_B$.
[†]That is, $\phi_{C^*}(\mathcal{D}) = 1 \iff \Lambda(\mathcal{D};\Theta_0) < C^*$.

Assumption 1 (Uniform consistency)**.** *The regression estimator used in Equation 2.8 is such that*

$$\sup_{\theta} |\widehat{\mathbb{E}}_{B'}[Z|\theta] - \mathbb{E}[Z|\theta]| \xrightarrow[B'\longrightarrow\infty]{a.s.} 0.$$

Examples of estimators that satisfy Assumption 1 include Bierens (1983); Hardle et al. (1984); Liero (1989); Girard et al. (2014).

The next theorem shows that the p-values obtained according to Algorithm 5 converge to the true p-values. Moreover, the power of the tests obtained using the estimates p-values converges to the power one would obtain if the true p-values could be computed.

Theorem 1. *Under Assumption 1, for every $\theta \in \Theta$,*

$$\widehat{p}(D;\Theta_0) \xrightarrow[B'\longrightarrow\infty]{a.s.} p(D;\Theta_0)$$

and

$$\mathbb{P}_{\mathcal{D},\mathcal{T}'|\theta}(\widehat{p}(\mathcal{D};\Theta_0) \leq \alpha) \xrightarrow{B'\longrightarrow\infty} \mathbb{P}_{\mathcal{D}|\theta}(p(\mathcal{D};\Theta_0) \leq \alpha).$$

The next corollary shows that as $B' \longrightarrow \infty$, the tests obtained using the p-values from Algorithm 5 have size α.

Corollary 1. *Under Assumption 1 and if F_θ is continuous for every $\theta \in \Theta$, then*

$$\sup_{\theta\in\Theta_0} \mathbb{P}_{\mathcal{D},\mathcal{T}'|\theta}(\widehat{p}(\mathcal{D};\Theta_0) \leq \alpha) \xrightarrow{B'\longrightarrow\infty} \alpha.$$

Under stronger assumptions about the regression method, it is also possible to derive rates of convergence for the estimated p-values.

Assumption 2 (Convergence rate of the regression estimator)**.** *The regression estimator is such that*

$$\sup_{\theta} |\widehat{\mathbb{E}}[Z|\theta] - \mathbb{E}[Z|\theta]| = O_P\left(\left(\frac{1}{B'}\right)^r\right).$$

for some $r > 0$.

Examples of regression estimators that satisfy Assumption 2 can be found in Stone (1982); Hardle et al. (1984); Donoho (1994); Yang et al. (2017).

Theorem 2. *Under Assumption 2,*

$$|p(D;\Theta_0) - \widehat{p}(D;\Theta_0)| = O_P\left(\left(\frac{1}{B'}\right)^r\right).$$

2.6.3 Power of `BFF`

In this section we provide convergence rates for `BFF` and show that its power relates to the loss function of Equation 2.10.

We assume that $G(\mathbf{x})$ is the marginal distribution of $F_\theta(\mathbf{x})$ with respect to $\pi(\theta)$. We here also assume that $\mathbf{x}$ contains all observations; that is, $\mathbf{X} = \mathcal{D}$. In this case, the denominator of the average odds is

$$\int_\Theta \mathbb{O}(\mathbf{x},\theta)d\pi(\theta) = \int_{\Theta_1} \frac{f(\mathbf{x}|\theta)}{g(\mathbf{x})}d\pi(\theta) = \int_\Theta \frac{f(\mathbf{x}|\theta)}{\int_\Theta f(\mathbf{x}|\theta)d\pi(\theta)}d\pi(\theta) = 1 \tag{2.11}$$

and therefore there is no need to estimate the denominator in Equation 2.6.

We also make the following assumptions:

Assumption 3 (Bounded odds and estimated odds)**.** *There exists* $0 < M, m < \infty$ *such that for every* $\theta \in \Theta$ *and* $\mathbf{x} \in \mathcal{X}$, $m \leq \mathbb{O}(\mathbf{x};\theta), \widehat{\mathbb{O}}(\mathbf{x};\theta) \leq M$.

Assumption 4 (Bounded second moment of odds estimation error)**.** *Let*

$$h(\theta) = \int (\mathbb{O}(\mathbf{x};\theta) - \widehat{\mathbb{O}}(\mathbf{x};\theta))^2 dG(\mathbf{x}).$$

There exists $M', m' > 0$ *such that* $h(\theta) \leq M'$ *and* $\int h(\theta)d\pi(\theta) > m'$.

Assumption 3 states that the odds and estimated odds are both bounded away from 0 and infinity, for all choice of parameters θ and features $\mathbf{x}$. Assumption 4 states that the second moment of the difference between the true and estimated odds is bounded away from 0 and infinity.

Finally, we assume that the CDF of the power function of the test based on τ is smooth in a Lipschitz sense:

Assumption 5 (Smooth power function)**.** *The cumulative distribution function of* $\tau(\mathcal{D};\theta_0)$, F_τ, *is Lipschitz with constant* C_L, *i.e., for every* $x_1, x_2 \in \mathbb{R}$, $|F_\tau(x_1) - F_\tau(x_2)| \leq C_L|x_1 - x_2|$.

With these assumptions, we can relate the odds loss with the probability that the outcome of `BFF` is different from the outcome of the test based on Bayes factor:

Theorem 3. *Let $\phi_\tau(\mathcal{D}) = \mathbb{I}\left(\tau(\mathcal{D};\theta_0) < c\right)$ and $\phi_{\widehat{\tau}_B}(\mathcal{D}) = \mathbb{I}\left(\widehat{\tau}_B(\mathcal{D};\theta_0) < c\right)$ be the testing procedures for testing $H_0 : \theta = \theta_0$ obtained using τ and $\widehat{\tau}_B$. Under Assumptions 3-5, there exists $K' > 0$ such that, for every $0 < \epsilon < 1$,*

$$\mathbb{P}_{\mathcal{D}|\theta,\mathcal{T}_B}(\phi_\tau(\mathcal{D}) \neq \phi_{\widehat{\tau}_B}(\mathcal{D})) \leq \frac{K' \cdot \sqrt{L(\widehat{\mathbb{O}}, \mathbb{O})}}{\epsilon} + \epsilon$$

Theorem 3 demonstrates that the probability that hypothesis tests based on the `BFF` statistic versus the Bayes factor lead to different conclusions is bounded by the `BFF` loss. This result is valuable because the `BFF` loss is easy to estimate in practice, and hence provides us with a practically useful metric. For instance, the `BFF` loss can serve as a natural criterion for selecting the "best" statistical model out of a set of candidate models with different classifiers, for tuning model hyperparameters, and for evaluating model fit.

Next, we provide rates of convergence of `BFF` to the test based on the Bayes factor. For that, we assume that the chosen probabilistic classifier has the following rate of convergence:

Assumption 6 (Convergence rate of the probabilistic classifier)**.** *The probabilistic classifier trained with $\mathcal{T}_B$, $\widehat{\mathbb{P}}(Y = 1|\mathbf{x}, \theta)$ is such that*

$$\mathbb{E}_{\mathcal{T}_B}\left[\int \left(\widehat{\mathbb{P}}(Y = 1|\mathbf{x}, \theta) - \mathbb{P}(Y = 1|\mathbf{x}, \theta)\right)^2 dH(\mathbf{x}, \theta)\right] = \mathcal{O}\left(B^{-\alpha/(\alpha+d)}\right),$$

for some $\alpha > 0$ and $d > 0$, where $H(\mathbf{x}, \theta)$ is a measure over $\mathcal{X} \times \Theta$.

Typically, α relates to the smoothness of $\mathbb{P}$, while d relates to the number of covariates of the classifier−in our case, the number of parameters plus the number of features. Below, we provide some examples where Assumption 6 holds, using well-established results for the convergence rates of commonly used regression estimators:

- Kpotufe (2011) shows that kNN estimators are adaptive to the intrinsic dimension d under certain conditions. When $\widehat{\mathbb{P}}$ is a kNN estimator with $\mathbb{P}$ in a class of Lipschitz continuous functions, Assumption 6 holds with $\alpha = 2$. More generally, with $\mathbb{P}$ in a

Hőlder space with parameter $0 < \beta \leq 1.5$, Assumption 6 holds with $\alpha = 2\beta$ (Győrfi et al. (2006); Ayano (2012)).

- Kpotufe and Garg (2013) show that under certain conditions, when $\widehat{p}$ is a kernel regression estimator with $\mathbb{P}$ in a class of Lipschitz continuous functions, Assumption 6 holds with $\alpha = 2$ and d the intrinsic dimension of the data. More generally, with $\mathbb{P}$ in a Hőlder space with parameter $0 < \beta \leq 1.5$, Assumption 6 holds with $\alpha = 2\beta$ (Győrfi et al. (2006)).

- When $\widehat{\mathbb{P}}$ is a local polynomial regression estimator with $\mathbb{P}$ in a Sobolev space with smoothness β, Assumption 6 holds with $\alpha = \beta$, where d is the manifold dimension (Bickel and Li (2007)).

Another noteworthy example outside of this category is the random forest regression estimator, for which Gao and Zhou (2020) have recently provided an improved bound of the order of $\mathcal{O}\left(B^{-\frac{1}{d+2}} \log(B)^{\frac{1}{d+2}}\right)$. More examples can be found in Győrfi et al. (2006), Tsybakov (2009) and Devroye et al. (2013).

We also assume that the density of the product measure $G \times \pi$ is bounded away from infinity.

Assumption 7 (Bounded density)**.** *$H(\mathbf{x}, \theta)$ dominates $H' := G \times \pi$, and the density of H' with respect to H, add "denoted by" h', is such that there exists $\gamma > 0$ with $h'(\mathbf{x}, \theta) < \gamma$, $\forall \mathbf{x} \in \mathcal{X}, \theta \in \Theta$.*

If the probabilistic has the convergence rate given by Assumption 6, the probability that hypothesis tests based on the `BFF` statistic versus the Bayes factor goes to zero with the rate given by the following theorem.

Theorem 4. *Under Assumptions 3-7, there exists $K'' > 0$ such that*

$$\mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_{\tau}(\mathcal{D}) \neq \phi_{\widehat{\tau}_B}(\mathcal{D})) \leq 2\sqrt{K''} B^{-\alpha/(4(\alpha+d))}.$$

Corollary 2 tells us that the power of the `BFF` test is close to the power of the exact Bayes Factor test. Because in the Neyman-Pearson setting the latter test is equivalent to the LRT, this implies that in this setting `BFF` converges to the most powerful test.

Corollary 2. *Under Assumptions 3-7, there exists $K'' > 0$ such that, for any $\theta \neq \theta_0$,*

$$\mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_{\widehat{\tau}_B}(\mathcal{D}) = 1) \geq \mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_{\tau}(\mathcal{D}) = 1) - 2\sqrt{K''}B^{-\alpha/(4(\alpha+d))}.$$

2.7 Sources of Error and a Practical Strategy

When constructing confidence sets using our inference machinery there are three separate sources of error that can affect confidence set validity and power:

e_1: Estimation error in learning the odds,

e_2: Numerical error when computing the maximization or integration (in the denominator for `ACORE` and `BFF` respectively), and

e_3: Estimation error in learning the critical values or the p-values.

e_1 and e_2 are the main drivers for determining the power of the downstream confidence sets. Even without odds estimation error ($e_1 = 0$), an incorrect estimation of the denominator of both test statistics can cause unwieldy fluctuations in the estimated values of the test statistic. At the same time, poorly estimated odds could be arbitrarily uninformative in identifying the correct region of the parameter space, even if one had access to the true values of the maximum or the integral at the denominator ($e_2 = 0$). As a note, the one exception is the case in which $n = 1$ and the marginal is used as reference distribution. Since the integral of the marginal in this case is 1 for any $\mathbf{X}$, using the `BFF` test statistic provides a clear advantage as it reduces the possible sources of error; see Section 2.9.5 for an application to galaxy images in such setting.

While power is tied to all three sources of error, validity is mostly determined by e_3, that is learning the critical values or p-values correctly. Using a quantile regression algorithm that can recover the critical value correctly as $B' \to \infty$ would construct a valid confidence set regardless of the quality of the test statistic approximation, as shown by Theorem 2.1. Instead, the critical values estimation is dependent on the smoothness of the estimated odds, which would impact the magnitude of the sample size B'. In addition, the critical or p-value estimation task is always lower-dimensional than the odds estimation task, as the simulated data $\mathcal{D} = (\mathbf{X}_1, ..., \mathbf{X}_n)$ are not included as features but are incorporated in the test statistic $\tau(\mathcal{D}; \theta)$.

In order to address all sources of uncertainty, a practitioner would need to decide on the following five key components:

(i) a probabilistic classifier for learning the odds,

(ii) a training sample of size B for learning the odds,

(iii) a sample budget M for integration or maximization,

(iv) a quantile regression algorithm to estimate critical values or a probabilistic classifier to estimate p-values, and

(v) a training sample size $B^{'}$ for estimating critical values.

We propose the following practical strategy to choose such components:

- Select (i) and (ii) according to which probabilistic classifier minimizes the cross-entropy loss on a held-out simulation set,
- Use our diagnostic tools in Section 2.4 to determine which combination of (iv) and (v) achieves the correct nominal coverage across the parameter space,
- Determine (iii) according to the current computational limitations, with the understanding that the higher M the better the approximation of the denominator of the test statistic will be.

One could also use the odds loss (2.10) proposed in Section 2.5 to select the probabilistic classifier for learning the odds. As we shall see in the Section 2.9, the proposed odds loss is sensitive to the value of the estimated odds and can yield very large positive or negative values across different training sample sizes B. On the other hand, cross entropy loss is more stable and most classifiers are trained to minimize the cross entropy by default.[‡]

[‡]One could also train deep neural networks to minimize the odds loss directly. See Section 5.1 for more details.

As a final note, throughout this chapter we set the proposal distribution over the parameter space $\pi(\theta)$ to be uniform. The proposal distribution determines the number of samples in specific regions of the parameter space Θ for both odds and critical values estimation (samples $\mathcal{T}_B$ and $\mathcal{T}'_{B'}$, respectively). The more samples from the neighborhood around the true value of the parameter (i.e., the parameter θ compatible with the observed data D), the better the approximation of the test statistic and critical value where the test statistic is the largest. Hence, a proposal distribution that places more probability mass around the true value of the parameter would result in confidence sets with a higher power and smaller sizes than when using a uniform proposal distribution. Conversely, a proposal distribution that places little probability mass around the true parameter value would result in lower power and larger size confidence sets, as quantile regression algorithms tend to be conservative with low sample sizes. One could use problem-specific prior knowledge or develop active learning strategies to inform the distribution of $\pi(\theta)$ to be concentrated at parameter values θ where the estimated test statistic $\widehat{\tau}(\mathrm{D};\theta)$ is largest.

2.8 Nuisance Parameters

In many applications, the parameter space can be decomposed as $\Theta = \Phi \times \Psi$, where Φ contains the parameters of interest and Ψ contains nuisance parameters not of immediate interest. Consider tests of the form $H_{0,\phi_0} : \phi = \phi_0$ versus $H_{1,\phi_0} : \phi \neq \phi_0$ for $\phi_0 \in \Phi$. Dealing with a large number of nuisance parameters ψ is challenging from a numerical perspective: the infima and suprema computation for `ACORE` can be numerically unwieldy (Zhu et al., 2020), while the the Monte Carlo estimates for `BFF` might be computationally infeasible (van den Boom et al., 2020).

In this work we approach nuisance parameters consistently with the definitions of the `ACORE` and `BFF` test statistics in equations (2.5) and (2.6), respectively. For `ACORE`, we propose to approximate these statistics via the hybrid resampling method, which is essentially a form of likelihood profiling (Chuang and Lai, 2000; Sen et al., 2009; Feldman, 2000). In particular, we first compute

$$\widehat{\psi}_\phi = \arg\max_{\psi \in \Psi} \prod_{i=1}^{n} \widehat{\mathbb{O}}\left(\mathbf{X}_i^{\mathrm{obs}}; (\phi, \psi)\right),$$

which is an approximation of the maximum likelihood estimate of ψ based on the observed data D at every value of the parameter of interest ϕ. Note that if $\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{x}) = \mathbb{P}(Y=1|\theta,\mathbf{x})$, then the ACORE statistic is equal to the profiled likelihood ratio as defined by Murphy and Van Der Vaart (2000) We then test $H_{0,\phi_0} : \phi = \phi_0$ by either computing the cutoff

$$\widehat{C}_{\phi_0} := \widehat{F}^{-1}_{\tau(\mathcal{D};(\phi_0,\widehat{\psi}_{\phi_0}))\big|(\phi_0,\widehat{\psi}_{\phi_0})}(\alpha)$$

where $\widehat{F}^{-1}$ is obtained via a quantile regression as in Algorithm 4 but using a training sample $\mathcal{T}'$ generated at fixed $\widehat{\psi}_{\phi_0}$, or by computing the p-value

$$\widehat{p}(D;\phi_0) := \widehat{\mathbb{P}}_{\mathcal{D}\big|T,(\phi_0,\widehat{\psi}_{\phi_0})}\left(\widehat{\tau}\left(\mathcal{D};\left(\phi_0,\widehat{\psi}_{\phi_0}\right)\right) < \widehat{\tau}\left(D;\left(\phi_0,\widehat{\psi}_{\phi_0}\right)\right)\right)$$

where $\widehat{\mathbb{P}}$ is obtained via a regression as in Algorithm 5 but with $\mathcal{T}'$ simulated at fixed $\widehat{\psi}_{\phi_0}$. This technique effectively reduces the parameter space to $\Theta' := \left\{\left(\phi, (\widehat{\psi}(\phi)\right) : \phi \in \Phi\right\}$, which has lower dimension than Θ. This makes it easier to estimate the key quantities of interest: By profiling, we are effectively computing our p-value regression or critical value quantile regression in a lower-dimensional space, the space of Θ'. Hybrid resampling does not always control α, but it is often a good approximation that leads to robust results (Aad et al., 2012; Qian et al., 2016). For BFF , we propose to integrate the odds over the nuisance parameters, by using a reference distribution $\pi(\psi)$; in other words, under the presence of nuisance parameter we define the following test statistic:

$$\tau(\mathcal{D};\phi_0) := \frac{\int_\Psi \prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};(\phi_0,\psi))d\pi(\psi)}{\int_\Theta \left(\prod_{i=1}^n \mathbb{O}(\mathbf{X}_i^{\text{obs}};\theta)\right)d\pi(\theta)}. \tag{2.12}$$

This is known in the high energy physics literature as the "hybrid Bayesian-frequentist treatment of nuisance parameters"(Cousins and Highland, 1992). Algorithm 6 details our construction of confidence sets in the presence of nuisance parameters for the ACORE test statistic; the algorithm for BFF would follow the same steps but integrate over the nuisance parameters (at Algorithm 6 line 6), passing only the parameters of interest as features for estimating the critical or p-value at every $\phi \in \Phi$. For both ACORE and BFF confidence sets we propose to use the diagnostic techniques in Section 2.4 to assess whether we obtain the nominal coverage or not in practice.

Algorithm 6 Construct confidence set for ϕ with coefficient $\gamma = 1 - \alpha$, in the presence of nuisance parameters ψ for the `ACORE` test statistic

Input: stochastic forward simulator F_θ; proposal distribution π over $\Theta = \Phi \times \Psi$; parameter p of Bernoulli distribution; sample size B (for estimating odds ratios); sample size B' (for estimating critical values or p-values); probabilistic classifier; observed data $D = \left\{\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}\right\}$; desired level $\alpha \in (0,1)$; number of parameter values at which to evaluate confidence set n_{grid}; confidence set estimation strategy `eval` (either `p-values` or `crit-values`)

Output: ϕ-values in confidence set

```
 1: // **Estimate odds**
 2: Generate labeled sample $\mathcal{T}_B$ according to Algorithm 1
 3: Apply probabilistic classifier to $\mathcal{T}_B$ to learn class posterior probabilities, $\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X})$,
    for all $\theta = (\phi,\psi) \in \Theta$ and $\mathbf{X} \in \mathcal{X}$
 4: Let the estimated odds $\widehat{\mathbb{O}}(\mathbf{X};\theta) = \frac{\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X})}{\widehat{\mathbb{P}}(Y=0|\theta,\mathbf{X})}$
 5: // **Profiling for** $\psi$
 6: Define $\widehat{\psi}_\phi \leftarrow \arg\max_{\psi\in\Psi} \widehat{\mathbb{O}}(\mathrm{D};(\phi,\psi))$ for every $\phi$
 7: // **Profiled test statistic** $\tau'$
 8: Define $\tau'(\mathcal{D};\phi) \leftarrow \tau(\mathcal{D};(\phi,\widehat{\psi}_\phi))$ for every $\phi$
 9: // **Find parameter set for which the test** $\delta_{\phi_0}$ **does not reject** $\phi = \phi_0$
10: $\mathrm{L}_\Phi \leftarrow$ lattice over $\Phi$ with $n_{\text{grid}}$ elements
11: Set $S \leftarrow \emptyset$
12: if eval == crit-values then
13:    Estimate critical value $\widehat{C}_\phi \leftarrow \widehat{F}^{-1}_{\tau'|\phi}(\alpha \mid \phi)$ for every $\phi$          (Algorithm 4)
14: else if eval == p-values then
15:    Estimate p-value $\widehat{p}(D;\phi) \leftarrow \widehat{\mathbb{E}}\left[\mathbb{I}\left(\widehat{\tau'}(\mathcal{D};\phi) < \tau'(D;\phi)\right) \mid \phi\right]$ for every $\phi$ (Algorithm 5)
16: end if
17: for $\phi_0 \in \mathrm{L}_\Phi$ do
18:    Compute the observed profiled test statistic $\widehat{\tau'}^{\text{obs}} \leftarrow \widehat{\tau}'(D;\phi_0) = \widehat{\tau}(D;(\phi_0,\widehat{\psi}_{\phi_0}))$
19:    if eval == crit-values then
20:       if $\widehat{\tau'}^{\text{obs}} > \widehat{C}_{\phi_0}$ then
21:          $S \leftarrow S \cup \{\phi_0\}$
22:       end if
23:    else if eval == p-values then
24:       if $\widehat{p}(D;\phi_0) > \alpha$ then
25:          $S \leftarrow S \cup \{\phi_0\}$
26:       end if
27:    end if
28: end for
29: return
```

2.9 Experiments and Applications

This section includes a series of examples of constructing confidence sets in a LFI setting spanning a range of different data and parameter settings. Section 2.9.1 considers two parametric models (Poisson and Gaussian mixture model) with a one-dimensional parameter and feature space. Section 2.9.1 showcases how ACORE and BFF scale to higher dimensional parameter spaces, using a high-dimensional Gaussian example. Sections 2.9.3 and 2.9.4 consider two examples inspired by particle collision experiments in high-energy physics. In particular, Section 2.9.4 showcases the use of ACORE and BFF for constructing confidence sets under the presence of nuisance parameters. Lastly, Section 2.9.5 illustrates an example in which the simulated data are 400-dimensional images.

2.9.1 One-Dimensional Parametric Models

We consider two examples where the true likelihood is known. In the first example, the forward simulator F_θ follows a $\text{Pois}(100 + \theta)$ distribution similar to the signal-background model in Section 2.9.3. In the second example, we consider a Gaussian mixture model (GMM) with two unit-variance Gaussians centered at $-\theta$ and θ, respectively. In both examples, $n = 10$, the proposal distribution r_Θ is a uniform distribution, and the reference distribution G is a normal distribution. Table 2.1 summarizes the set-up.

	Poisson Example	*GMM Example*
r_Θ	$\text{Unif}(0, 20)$	$\text{Unif}(0, 10)$
F_θ	$\text{Pois}(100 + \theta)$	$\frac{1}{2}\mathcal{N}(-\theta, 1) + \frac{1}{2}\mathcal{N}(\theta, 1)$
G	$\mathcal{N}(110, 15^2)$	$\mathcal{N}(0, 5^2)$
True θ	$\theta_0 = 10$	$\theta_0 = 5$

Table 2.1: Set-up for the two toy examples.

First we investigate how the power of ACORE and BFF and the size of the derived confidence sets depend on the performance of the classifier used in the odds ratio estimation (Section 2.2) and the use of critical or p-values in constructing confidence sets (Section 2.3). We consider three classifiers for odds estimation: multilayer perceptron (MLP), nearest neighbor (NN) and quadratic discriminant analysis (QDA). We use gradient boosted quantile regression for critical value estimation and a MLP classifier for p-values estimation,

Poisson Model

CI Method	*Test Statistic*	*B*	*Classifier*	*Cross Entropy*	*Power*	*CI Size*	*Coverage*
Crit. Values	`ACORE`	100	MLP	0.87 ± 0.27	0.24	75.9 ± 19.3	0.91
			NN	0.76 ± 0.15	0.29	71.6 ± 19.7	0.91
			QDA	0.66 ± 0.02	0.41	60.0 ± 15.6	0.90
	`BFF`	100	MLP	0.87 ± 0.27	0.18	82.2 ± 17.6	0.94
			NN	0.76 ± 0.15	0.27	72.9 ± 20.9	0.87
			QDA	0.66 ± 0.02	0.49	52.4 ± 10.6	0.91
	`ACORE`	500	MLP	0.69 ± 0.01	0.35	65.9 ± 20.4	0.91
			NN	0.67 ± 0.01	0.38	62.9 ± 15.8	0.93
			QDA	0.64 ± 0.01	0.47	54.2 ± 9.4	0.94
	`BFF`	500	MLP	0.69 ± 0.01	0.35	66.0 ± 19.2	0.92
			NN	0.67 ± 0.01	0.42	58.9 ± 14.2	0.87
			QDA	**0.64 ± 0.01**	**0.53**	**47.9 ± 7.0**	0.87
	`ACORE`	1000	MLP	0.69 ± 0.01	0.37	63.3 ± 19.7	0.91
			NN	0.66 ± 0.01	0.44	56.9 ± 15.9	0.89
			QDA	0.64 ± 0.01	0.50	51.3 ± 7.7	0.91
	`BFF`	1000	MLP	0.69 ± 0.01	0.36	64.4 ± 20.0	0.93
			NN	0.66 ± 0.01	0.42	58.7 ± 12.3	0.93
			QDA	**0.64 ± 0.01**	**0.53**	**48.4 ± 5.7**	0.91
P-Values	`ACORE`	100	MLP	0.87 ± 0.27	0.26	74.8 ± 21.0	0.93
			NN	0.76 ± 0.15	0.22	78.3 ± 19.9	0.96
			QDA	0.66 ± 0.02	0.41	59.6 ± 17.4	0.91
	`BFF`	100	MLP	0.87 ± 0.27	0.14	86.4 ± 18.5	1.00
			NN	0.76 ± 0.15	0.27	74.0 ± 17.4	0.97
			QDA	0.66 ± 0.02	0.45	56.4 ± 12.0	0.93
	`ACORE`	500	MLP	0.69 ± 0.01	0.33	67.6 ± 21.9	0.89
			NN	0.67 ± 0.01	0.37	63.6 ± 18.7	0.84
			QDA	0.64 ± 0.01	0.50	51.3 ± 10.6	0.84
	`BFF`	500	MLP	0.69 ± 0.01	0.25	75.7 ± 21.4	0.99
			NN	0.67 ± 0.01	0.38	63.0 ± 15.5	0.91
			QDA	0.64 ± 0.01	0.53	48.3 ± 7.0	0.91
	`ACORE`	1000	MLP	0.69 ± 0.01	0.42	58.4 ± 21.3	0.88
			NN	0.66 ± 0.01	0.39	62.1 ± 17.6	0.88
			QDA	0.64 ± 0.01	0.49	51.8 ± 10.3	0.90
	`BFF`	1000	MLP	0.69 ± 0.01	0.32	68.5 ± 21.8	0.96
			NN	0.66 ± 0.01	0.41	59.5 ± 15.0	0.92
			QDA	**0.64 ± 0.01**	**0.54**	**47.3 ± 5.8**	0.89
Exact LR	-	-	-	**0.64 ± 0.01**	**0.54**	**45.0 ± 4.9**	**0.90**

Table 2.2: Results for Poisson example. The table shows the cross entropy loss, power (averaged over θ), size and coverage of confidence sets for different values of B and for different classifiers. These results are based on 100 repetitions; the numbers represent the mean and one standard deviation. The best results in each setting are marked in bold-faced; we see that the classifier with the lowest cross entropy loss is linked with the highest average power and the smallest confidence set. All confidence sets are valid at the nominal level.

GMM Model

CI Method	*Test Statistic*	*B*	*Classifier*	*Cross Entropy*	*Power*	*CI Size*	*Coverage*
Crit. Values	ACORE	100	MLP	0.39 ± 0.03	0.88	14.1 ± 4.7	0.87
			NN	0.81 ± 0.31	0.42	58.4 ± 23.3	0.91
			QDA	0.64 ± 0.02	0.15	85.3 ± 21.1	0.88
	BFF	100	MLP	0.39 ± 0.03	0.88	13.5 ± 3.5	0.85
			NN	0.81 ± 0.31	0.50	50.9 ± 24.0	0.86
			QDA	0.64 ± 0.02	0.14	86.1 ± 19.9	0.91
	ACORE	500	MLP	**0.35 ± 0.01**	**0.90**	**12.1 ± 2.4**	0.91
			NN	0.45 ± 0.05	0.57	44.3 ± 24.1	0.95
			QDA	0.62 ± 0.01	0.15	84.9 ± 19.9	0.92
	BFF	500	MLP	**0.35 ± 0.01**	**0.90**	**11.7 ± 1.9**	0.91
			NN	0.45 ± 0.05	0.72	29.7 ± 16.8	0.90
			QDA	0.62 ± 0.01	0.12	88.0 ± 15.8	0.91
	ACORE	1000	MLP	**0.35 ± 0.01**	**0.90**	**12.1 ± 2.5**	0.92
			NN	0.41 ± 0.02	0.77	24.9 ± 15.9	0.84
			QDA	0.62 ± 0.01	0.12	88.1 ± 18.0	0.93
	BFF	1000	MLP	**0.35 ± 0.01**	**0.90**	**11.6 ± 2.2**	0.86
			NN	0.41 ± 0.02	0.83	18.5 ± 9.1	0.87
			QDA	0.62 ± 0.01	0.12	88.0 ± 15.7	0.94
P-Values	ACORE	100	MLP	0.39 ± 0.03	0.88	13.7 ± 4.0	0.89
			NN	0.81 ± 0.31	0.51	50.2 ± 23.6	0.94
			QDA	0.64 ± 0.02	0.17	83.6 ± 21.8	0.90
	BFF	100	MLP	0.39 ± 0.03	0.88	13.6 ± 4.4	0.88
			NN	0.81 ± 0.31	0.53	48.5 ± 22.0	0.89
			QDA	0.64 ± 0.02	0.14	85.8 ± 20.7	0.87
	ACORE	500	MLP	**0.35 ± 0.01**	**0.90**	**11.9 ± 2.8**	0.90
			NN	0.45 ± 0.05	0.74	27.1 ± 17.4	0.92
			QDA	0.62 ± 0.01	0.15	84.6 ± 19.3	0.88
	BFF	500	MLP	**0.35 ± 0.01**	**0.90**	**12.0 ± 2.3**	0.87
			NN	0.45 ± 0.05	0.74	27.2 ± 15.0	0.92
			QDA	0.62 ± 0.01	0.12	88.1 ± 18.2	0.91
	ACORE	1000	MLP	**0.35 ± 0.01**	**0.90**	**11.4 ± 2.2**	0.94
			NN	0.41 ± 0.02	0.84	17.9 ± 9.4	0.95
			QDA	0.62 ± 0.01	0.18	82.3 ± 22.9	0.89
	BFF	1000	MLP	**0.35 ± 0.01**	**0.90**	**11.7 ± 2.0**	0.85
			NN	0.41 ± 0.02	0.83	18.5 ± 9.6	0.96
			QDA	0.62 ± 0.01	0.10	89.6 ± 16.6	0.92
Exact LR	-	-	-	**0.35 ± 0.01**	**0.92**	**9.5 ± 2.0**	**0.90**

Table 2.3: Results for GMM example. The table shows the cross entropy loss, power (averaged over θ), size and coverage of confidence sets for different values of B and for different classifiers. These results are based on 100 repetitions; the numbers represent the mean and one standard deviation. The best results in each setting are marked in bold-faced; we see that the classifier with the lowest cross entropy loss is linked with the highest average power and the smallest confidence set. All confidence sets are valid at the nominal level.

with $B^{'} = 5000$. For different values of B (sample size for estimating odds ratios), we compute the binary cross entropy (a measure of classifier performance), the power as a function of θ, the size of the constructed confidence set as a proportion of the parameter space Θ and the coverage of the confidence set. Tables 2.9.1 and 2.9.1 summarize the results based on 100 repetitions. Our results show that we for all cases achieve results in line with the nominal confidence level.[§] The last row of the table shows the best attainable cross entropy loss (see Section 2.7 for more details), the confidence set size and power for the true likelihood function.

For each setting with fixed B, the best classifier according to cross entropy loss achieves the highest power and the smallest confidence set.[¶] This trend is consistent in both `ACORE` and `BFF` across confidence sets constructed using critical values and p-values. Moreover, as B increases, the best values (marked in bold-faced) get closer to those of the true likelihood (marked in red). The cross-entropy loss is easy to compute in practice. Our results indicate that minimizing the cross-entropy loss is a good rule of thumb for achieving inference results with desirable statistical properties. In addition, for these one-dimensional examples, `BFF` seem to consistently achieve a slightly higher power than `ACORE`, although using the same computational budget for maximization and integration ($M = 1000$).

Next we illustrate our goodness-of-fit procedure (Section 2.4) for checking the coverage of the constructed confidence sets across the parameter space Θ with the `ACORE` test statistic. To pass our goodness-of-fit test, we require the nominal coverage to be within two standard deviations of the estimated coverage for all parameter values. Our goodness-of-fit procedure shown in Figure 2.3 uses a set $\mathcal{T}^{''}_{B''}$ with size $B^{''} = 250$ and logistic regression to generate prediction bands. Figure 2.3 shows the estimated coverage with logistic regression for both the Poisson and GMM example with $B = 1000$ (learning odds via a QDA and MLP classifier respectively), over three different values of $B^{'}$ (the training sample size for estimating the critical value C via gradient boosted quantile regression). As expected (Theorem 2.1), the estimated coverage gets closer to the nominal 90% confidence level as B' increases. We can use these diagnostic plots to choose B'. For instance, here $B' = 500$ in the Poisson example

[§]The 95% CI of a binomial distribution with probability $p = 0.9$ over 100 repetitions is in fact $[0.84, 0.95]$. This interval includes the observed coverages listed in Tables 2.9.1 and 2.9.1.

[¶]In traditional settings, high power has been shown to lead to a small expected interval size under certain distributional assumptions (Pratt, 1961; Ghosh, 1961).

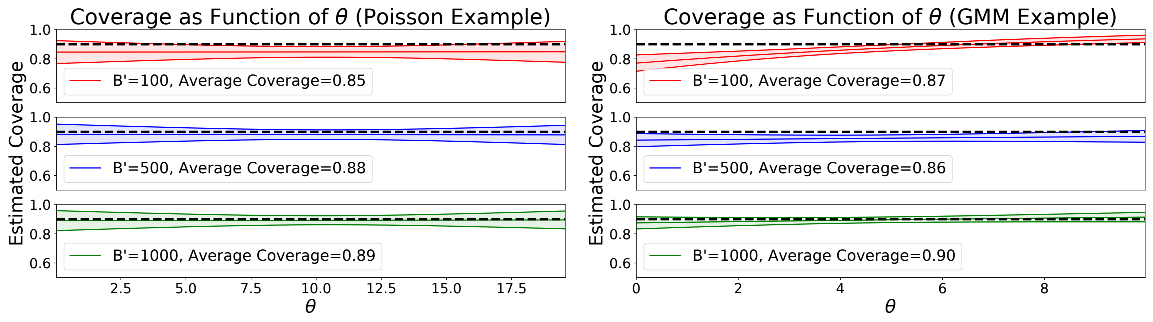


Figure 2.3: Estimated coverage as a function of θ in the Poisson example (left) and GMM example (right) for ACORE with different values of B'. The mean and one standard deviation prediction intervals are estimated via logistic regression. Our diagnostics show that $B' = 500$ is large enough to achieve the nominal confidence level $1 - \alpha = 0.9$ in the Poisson example, while a $B' = 1000$ is necessary in the GMM example. (We here use $n = 10$, a quantile gradient boosted trees for critical value estimation, and a QDA and MLP classifier in the Poisson and GMM example respectively.

is large enough for ACORE to achieve good coverage, while the GMM example requires $B^{'} = 1000$. In addition, the coverage for the GMM example shows a noticeable tilt in the prediction bands for $B' = 100$ and 500. However, as B' increases, the estimation of critical values becomes more precise and the estimated confidence intervals pass our goodness-of-fit diagnostic at, for example, $B' = 1000$.

Next we compare our goodness-of-fit diagnostic with diagnostics obtained via standard Monte Carlo sampling. Figure 2.4 shows the MC coverage as a function of θ for the Poisson example (left) and the Gaussian mixture model example (right). In both cases 100 MC samples are drawn at 100 parameter values chosen uniformly. The empirical ACORE coverage is computed over the MC samples at each chosen θ. This MC procedure is expensive: it uses a total of $10,000$ simulations, which is 40 times the number used in our goodness-of-fit procedure. The observed coverage of the Poisson example (Figure 2.4, left) indicates that $B' = 500$ is sufficient to achieve the nominal coverage of 90%. For the Gaussian mixture model example (Figure 2.4, right), we detect undercoverage for very small values of θ. This discrepancy is due to the fact that, at $\theta = 0$, the mixture collapses into a single Gaussian, structurally different from the GMM at any other $\theta > 0$ and closer to the $\mathcal{N}(0, 5^2)$ reference distribution.

Sections 2.10.1 and 2.10.2 include a comparison between ACORE and Monte Carlo Gaussian Process (MC GP) interpolation Frate et al. (2017) and calibrated neural nets classifiers (CARL, Cranmer et al. 2015), respectively. Our results show that MC-based GP interpolation provides a better approximation of the likelihood ratio when the simulated data are approximately Gaussian (as in the Poisson example). However, when the parametric assumptions are not valid (as in the GMM example), MC-based GP fails to approximate the likelihood ratio regardless of the number of available simulations. For both examples, CARL leads to lower power and larger confidence intervals than ACORE. See Tables 2.7, 2.8, 2.9 and 2.10 for details.

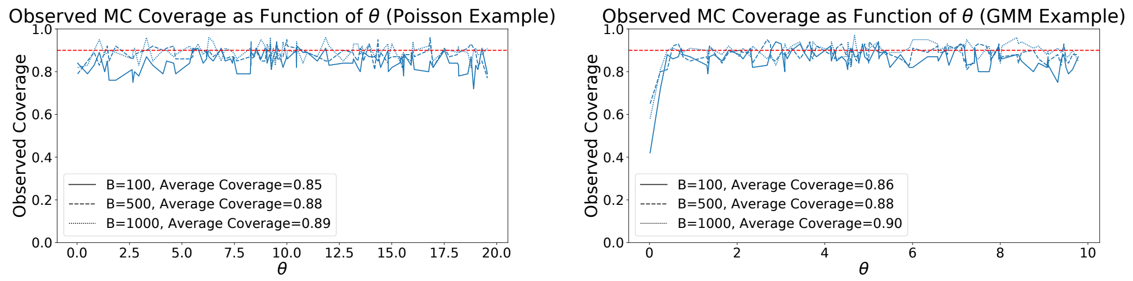


Figure 2.4: Observed ACORE coverage across the parameter space for the Poisson example (left) and the Gaussian mixture model example (right). The coverage is computed with Monte Carlo samples of size 100, each sampled at a θ chosen uniformly over the parameter space. Odds ratios are computed with a QDA classifier for the Poisson example, and an MLP classifier for the GMM example (as in Figure 2.3). We observe undercoverage at small θ for the GMM (right) due to the mixture collapsing into a single Gaussian as $\theta \to 0$.

2.9.2 High-Dimensional Gaussian Model

In this section we analyse the behavior of ACORE and BFF confidence sets as the dimension d of the parameter space increases using a Gaussian parametric model. Let $\mathbf{X}_1, \ldots, \mathbf{X}_n \sim N(\theta, I_d)$, where I_d is the d-dimensional identity matrix and $\theta \in \mathbb{R}^d$ is an unknown parameter. Given the sample mean $\bar{\mathbf{X}}$, the likelihood ratio for testing the null hypothesis $H_0 : \theta = \theta_0$ vs. $H_1 : \theta \neq \theta_0$ is equal to:

$$\mathrm{LR}(\mathbf{X}_1, \ldots, \mathbf{X}_n; \theta_0) = 2 \log \frac{N(\bar{\mathbf{X}}; \bar{\mathbf{X}}, n^{-1} I_d)}{N(\bar{\mathbf{X}}; \theta_0, n^{-1} I_d)}, \tag{2.13}$$

which we use as a benchmark for ACORE throughout this section. As a benchmark for BFF we also include the exact Bayes factor, for which we compute critical value via Monte Carlo sampling.

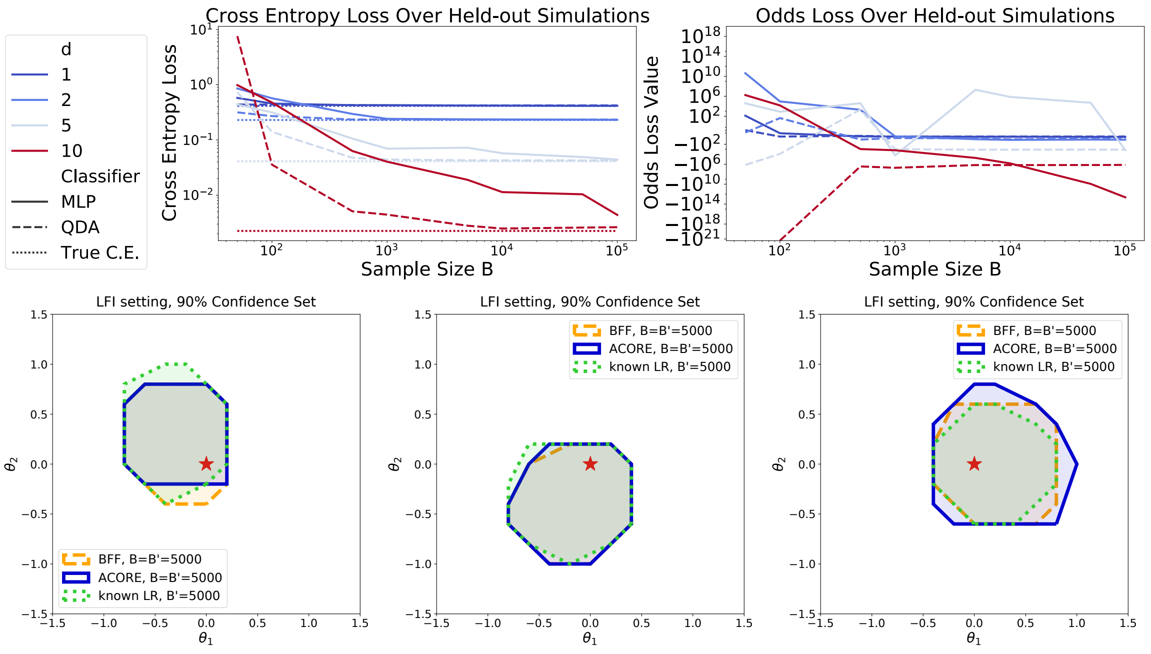


Figure 2.5: *Top*: Cross entropy loss and odds loss as a function of the sample size B for quadratic discriminant analysis classifier (QDA) and multilayer perceptron (MLP), for dimensions $d = 1, 2, 5$ and 10. As we increase B, the cross entropy loss decreases, suggesting a more accurate odds estimation. QDA achieves the lowerst cross-entropy loss among the classifiers we considered (of which MLP is an example, shown here for comparison). The odds loss also show a decreasing trend as the sample size B increases, although significantly more unstable. *Bottom*: When $d = 2$, BFF and ACORE confidence sets are of a similar size to those constructed using the known LR. In these example confidence sets, the true parameter is $\theta_0 = (0, 0)$ (indicated with a red star), $n = 10$ observations, $\alpha = 0.1$, $B = B' = 5000$ samples for BFF and ACORE; all confidence sets are constructed using the critical value estimation (Algorithm 4).

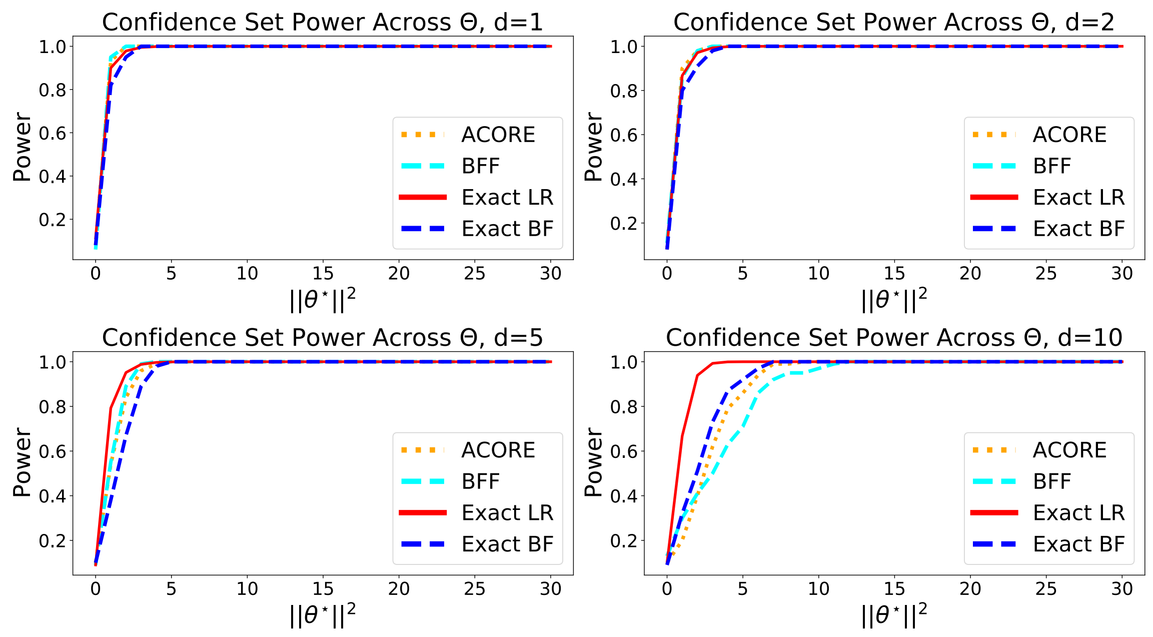
Firstly, as highlighted in our practical strategy in Section 2.7, we select the probabilistic classifier and sample size B according to the lowest cross entropy loss value on a held-out set. In Figure 2.5 (top) we show the cross entropy and odds loss for two probabilistic classifiers, a multilayer perceptron (MLP) and quadratic discriminant analysis (QDA), across dimensions $d = 1, 2, 5$ and 10. We include both MLP and QDA to show the difference in terms of cross entropy loss performance between the best performing classifier (QDA) against another classifier; for instance, in dimension $d = 10$, QDA achieves the same cross entropy loss as MLP with a factor of 200 fewer training samples. We also include the results for the odds loss, which mostly leads to similar conclusions as the cross entropy loss but is significantly more unstable across the sample size B. Finally, note that in some cases increasing B leads to diminishing marginal returns; identifying where the cross entropy loss plateaus can inform the choice of sample size B.

We then investigate `BFF`, `ACORE` and exact likelihood ratio confidence sets in low dimensions, using critical value estimation for confidence set construction in all three cases. Figure 2.5 (bottom) shows three realizations of the confidence sets for the true parameter $\theta_0 = (0, 0)$ with an observed sample size $n = 10$, using $B = B' = 5000$, $M = 1000$ and gradient boosted quantile regression trees. We see that `BFF` and `ACORE` confidence sets are similar in size and variability to the likelihood ratio ones, pointing at both the odds estimation error as well as maximization and integration numerical error to be small in dimension $d = 2$.

Next, we explore the power of `ACORE` , `BFF` and known likelihood confidence sets as the dimension d increases, fixing the true parameter $\theta_0 = \mathbf{0}$. For the likelihood ratio, we notice that in Equation 2.13, $\mathrm{LR}(\mathbf{X}_1, \ldots, \mathbf{X}_n; \boldsymbol{\theta}_0) \sim \chi^2_d$. Therefore, as $d \to \infty$ or $n\|\boldsymbol{\theta}\|^2 \to \infty$, the power of the likelihood ratio test (LRT) converges to

$$\Phi\left(\frac{d + n\|\boldsymbol{\theta}\|^2 - C_{\alpha,d}}{\sqrt{2(d + 2n\|\boldsymbol{\theta}\|^2)}}\right) \tag{2.14}$$

where $C_{\alpha,d}$ is the upper α quantile of a χ^2_d distribution and $\Phi(\cdot)$ is the standard normal CDF. As it is challenging to visualize power in higher dimensions, in this experiment we first apply the Neyman inversion only on parameter values θ with equal coordinate values; specifically on parameters $\theta^\star = \left(\frac{c}{\sqrt{d}}, \ldots, \frac{c}{\sqrt{d}}\right)$, for different values of the parameter c such that the L_2 norm $\|\theta^\star\|_2^2 \in [0, 30]$. We then compute the power by counting the proportion of times each parameter was included in the generated confidence set. Figure 2.6 shows the power as a function of the L_2 norm of the point considered in the Neyman inversion (which is the distance from the true parameter) for the `ACORE` , `BFF` and known likelihood confidence sets in dimensions $d = 1, 2, 5, 10$. Power and coverage are computed over 100 repetitions using a sample size $n = 10$. We use $B = 10000$, $B' = 10000$, $M = 10000$, a



	d=1	d=2	d=5	d=10
Coverage of `ACORE`	0.92 ± 0.03	0.92 ± 0.03	0.90 ± 0.03	0.90 ± 0.03
Coverage of `BFF`	0.94 ± 0.06	0.89 ± 0.03	0.91 ± 0.03	0.91 ± 0.03

Figure 2.6: In the LFI setting, where constructing confidence sets requires the estimation of both odds and critical values, the curse of dimensionality significantly affects the confidence set power. Here, we test $H_0 : \theta = \mathbf{0}$ at level $\alpha = 0.1$ for varying values of the true L_2-norm $\|\theta\|^2$ of the unknown parameter (see text), with $n = 10$ observations, results averaged over 100 trials, and using $B = 10000$ and $B' = 10000$ samples to train the probabilistic classifier and quantile regression in our method. Given the same computational budget for maximization and integration respectively ($M = 10000$), both `ACORE` and `BFF` lose power at higher dimension, with the effect being more pronounced at $d = 10$.

QDA classifier for odds estimation and a gradient boosted quantile regression algorithm for critical value estimation. We see that all three methods construct valid confidence sets with very similar power at $d = 1$ and $d = 2$. Although `ACORE` and `BFF` confidence sets have comparable power to their exact counterparts in low dimensions, the power degrades as d grows. Nevertheless, confidence sets constructed with both test statistics consistently achieve the nominal level $1 - \alpha$ coverage[‖]. This suggest that in higher dimensions the primary challenge in our framework is ensuring that confidence sets have high power. Note that `ACORE` power degrades slower than `BFF` as a function of d, and that the same phenomena can be observed for the exact LR and BF respectively as well. This is not surprising: in

[‖]The coverage falls within or above expected variation for 100 repetition, which is in the range [84, 95].

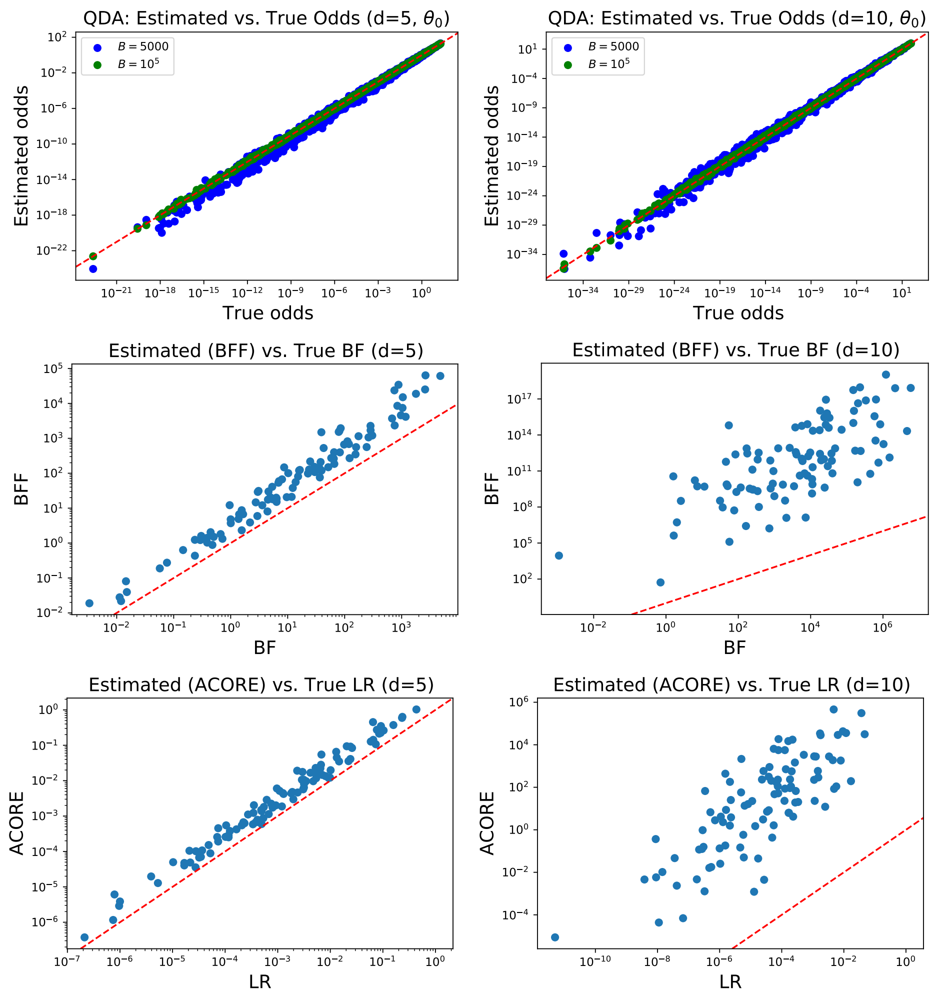


Figure 2.7: *Top*: Odds classifiers trained on B samples, evaluated on 1000 test samples, showing how QDA estimates the odds well for higher dimension d even with smaller sample sizes B. *Center and Bottom*: `BFF` and `ACORE` test statistics at $d = 5$ and $d = 10$ using the exact odds but using a budget of $M = 30000$ samples to estimate the denominator of both test statistics. As d grows, the numerical estimation error becomes larger, leading to an under-estimation of the denominator of both test statistics, hence to inflated values of `ACORE` and `BFF` .

a setting with a Gaussian likelihood and known variance, the exact LR is the uniformly most powerful unbiased tests for the Neyman construction null hypothesis, i.e., $H_0 : \theta = \theta_0$ against $H_A : \theta \neq \theta_0$ (Birkes, 1990).

To pinpoint the cause of the degradation in power in high dimension for `ACORE` and `BFF`, we separate the error in estimating the odds from the numerical error in the maximization or integration step in the test statistic (errors e_1 and e_2 in Section 2.7). Figure 2.7 (top panel) shows how QDA estimates odds well at both $d = 5$ and $d = 10$ (even at lower sample sizes B), at least at the true parameter value θ_0. Combining this with the cross entropy performance in Figure 2.5, it is safe to assume the odds estimation error is likely small. To isolate the numerical error, Figure 2.7 (bottom panel) shows the estimated `ACORE` and `BFF` using the analytical odds function at $d = 5$ and $d = 10$. Even a large budget of $M = 10000$ underestimate both the odds maximum and the integrated odds across the parameter space, resulting in an over-estimation of the `ACORE` and `BFF` test statistics. Hence, we foresee both `ACORE` and `BFF` benefit for more efficient numerical approaches; see Section 5.3 for a review of more efficient approaches for both maximization and integration.

2.9.3 Signal Detection in High-Energy Physics

In this section we construct confidence sets in a model described in Rolke et al. (2005) and Sen et al. (2009) for a high energy physics (HEP) experiment. In this model, particle collision events are counted under the presence of a background process b. The goal is to assess the intensity ν of a signal (i.e., an event which is not part of the background process). The observed data D consist of $n = 10$ realizations of $\mathbf{X} = (N, M)$, where $N \sim \text{Pois}(b + \nu)$ is the number of events in the signal region, and $M \sim \text{Pois}(b)$ is the number of events in the background (control) region. (We use a uniform proposal distribution r_Θ and a Gaussian reference distribution G.) This model is a simplified version of a real particle physics experiment where the true likelihood function is not known.

Figure 2.8 illustrates the role of B, B', and our goodness-of-fit procedure when estimating confidence sets. In the top left panel, we use the true LR statistic to show that, even if the LR is available, estimating the critical value C well still matters. Our goodness-of-fit diagnostic provides a principled way of choosing the best quantile regression (QR) method and the best sample size B' for estimating C. In this example, random forest QR does not pass our goodness-of-fit test; it also leads to a confidence region quite different from the exact one. Deep QR, which passes our test, gives a more accurate region estimate. In the top right panel, we use `ACORE` to estimate both the odds ratio and the critical value C (this is the LFI setting). If we choose B by identifying when the cross entropy loss levels off, we would choose $B = 100000$. Decreasing B leads to a worse cross-entropy loss and, as

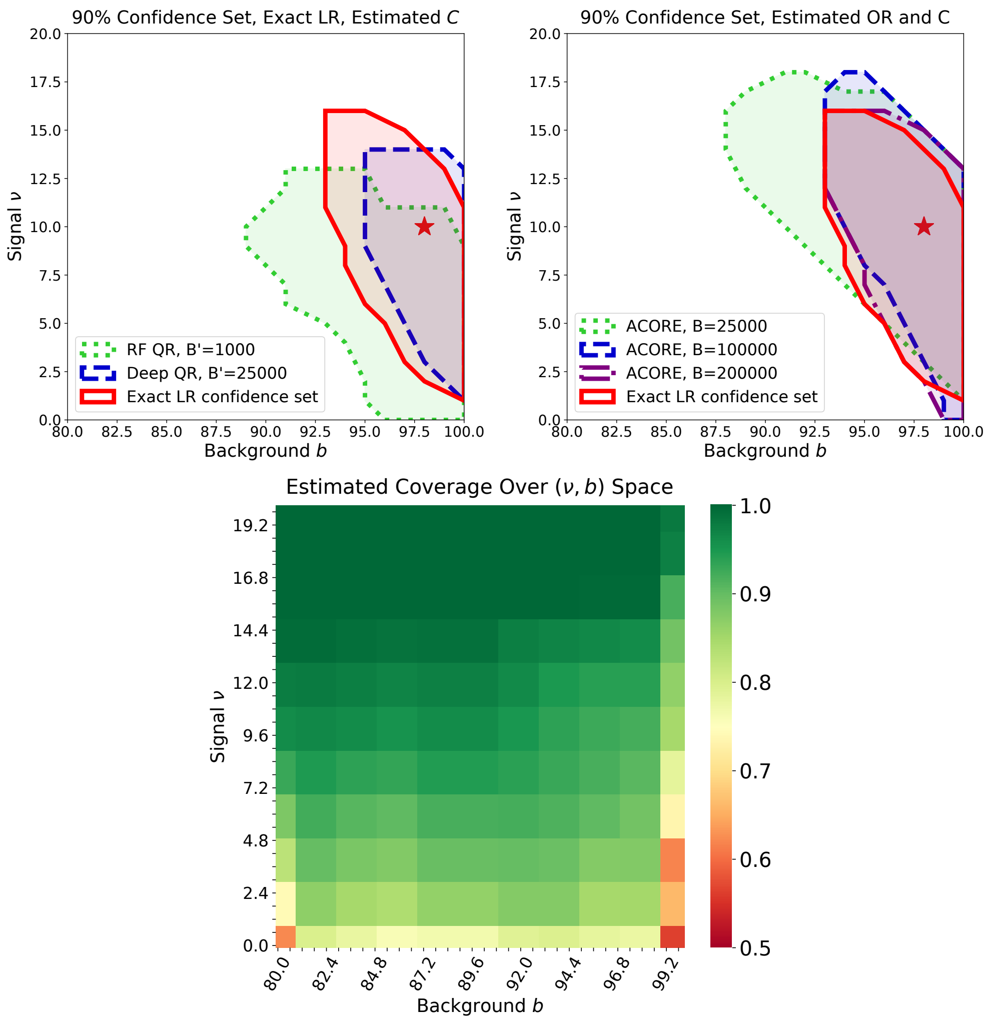


Figure 2.8: *Top Left*: 90% confidence sets computed with the exact likelihood ratio statistic. Estimating critical values can however be challenging, as highlighted by the differences in the results for two different quantile regression (QR) algorithms and sample sizes: Random Forest QR at $B' = 1000$ (green dotted) versus Deep QR at $B' = 25000$ (blue dashed). Our goodness-of-fit procedure can be used to select the best method in a principled way. (The red contour shows the exact LR confidence set, and the red star is at the true parameter setting.) *Top Right*: 90% confidence sets when estimating both odds ratios and critical values. This is the LFI setting. Our proposed strategy for choosing the different components of our inference machinery selects a 5-layer deep neural network with $B = 100000$; this yields a confidence set (dashed blue) close to the exact LR set (solid red). Increasing B does not show a noticeable improvement (dash-dotted purple), whereas decreasing B makes estimates worse (dotted green). *Bottom*: Heat map of the estimated coverage for a confidence set that did *not* pass our goodness-of-fit diagnostic. The overall coverage of the confidence set is correct (91.8% vs. the 90% nominal confidence level), but the set clearly undercovers in low-signal and high-background regions.

the figure shows, also a larger confidence region. Increasing B beyond our selected sample size does not lead to substantial gains. The bottom panel illustrates how our goodness-of-fit procedure can be used to identify regions in parameter space where a constructed confidence set is not valid. The heat map refers to an example which did not pass our goodness-of-fit procedure. While the overall (marginal) coverage is at the right value, our diagnostic procedure (for estimating coverage as a function of ν and b) is able to identify undercoverage in low-signal and high-background regions. That is, for a valid confidence set, one needs to better estimate the critical value C by, e.g., using a different quantile regression estimator or by increasing B' (either uniformly over the parameter space or by an active learning scheme which increases the number of simulations at parameter settings where one undercovers).

https://www.overleaf.com/project/5fd2b66ef24b695c2baa316f Figure 2.9 illustrates the two steps in identifying the four components of our inference machinery. We first use a validation set of $5,000$ simulations to determine which probabilistic classifier and training sample size B minimize the cross entropy loss. Figure 2.9 (left) shows the cross entropy loss of the best four classifiers as function of B. The minimum is achieved by a 5-layer deep neural network (DNN) at $B = 100,000$ with a cross entropy loss of 58.509×10^{-2}, closely followed by QDA with 58.512×10^{-2} at $B = 50,000$. Given how similar the loss values are, we select both classifiers to follow-up on. In Figure 2.9 (right), the "estimated correct coverage" represents the proportion of the parameter space that passes our diagnostic procedure. The lowest B' with correct coverage is achieved by the five-layer DNN classifier (for estimating odds ratios) at $B' = 25,000$ with critical values estimated via a two-layer deep quantile regression algorithm. None of the quantile regression algorithms pass a diagnostic test with a nominal coverage of 90% at the one standard deviation level when using the QDA classifier; we therefore do not include QDA in this example. Given the above, the following components were chosen: (i) a five-layer DNN for learning odds ratios, (ii) $B = 100,000$, (iii) a two-layer deep quantile regression for estimating critical values, and (iv) $B' = 25,000$.

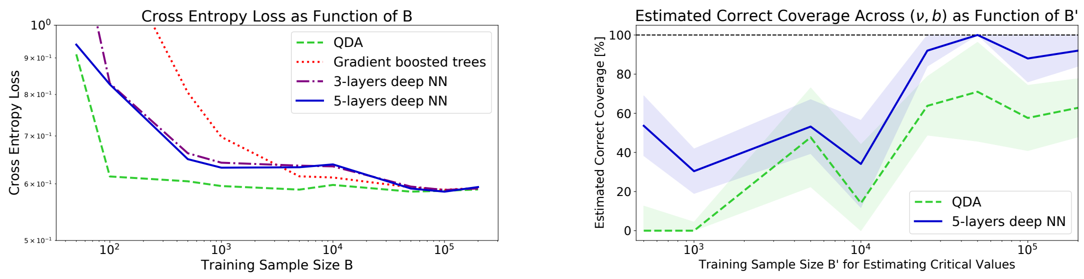


Figure 2.9: *Left:* The cross entropy loss of the best four classifiers, shown as a function of B. In order of increasing loss: 5-layer DNN ([512, 256, 64, 32, 32] neurons, ReLu activations), QDA classifier, 3-layer DNN ([64, 32, 32] neurons, ReLu activations) and gradient boosted trees (1000 trees with maximum depth 5). Because the first two classifiers (the 5-layer DNN and QDA) achieve a very similar minimum loss, we consider both classifiers in the follow-up step. *Right*: Proportion of the (ν, b) parameter space where the best two classifiers pass our goodness-of-fit procedure with a nominal coverage of 90%. Both the mean value curves and the $\pm$ one standard deviation prediction bands are computed via logistic regression. Critical values are estimated via a two-layer deep quantile regression ([64,64] neurons, ReLu activations), which passed the diagnostic at the lowest sample size ($B' = 25,000$, with the 5-layers DNN). Based on the results, we choose the 5-layer DNN with $B' = 25,000$.

2.9.4 3D Mixtures in High-Energy Physics

In this section we illustrate the construction of confidence sets under the presence of nuisance parameters using a HEP synthetic model from De Castro and Dorigo (2019). In this model, particle collision events are counted under the presence of a background process b. The goal is to identify the intensity of a signal s, i.e. an event which is not part of the background process. The simulated data $\mathbf{x} = (x_0, x_1, x_2) \in \mathbb{R}^3$ come from a 3D mixture model, which is defined as follows:

$$p(\mathbf{x}|s, r, \lambda, b) = \frac{b}{b+s} f_b(\mathbf{x}|r, \lambda) + \frac{s}{s+b} f_s(\mathbf{x}), \tag{2.15}$$

where:

$$f_b(\mathbf{x}|r,\lambda) = \mathcal{N}\left((x_0, x_1)|(2+r,0), \begin{bmatrix} 5 & 0 \\ 0 & 9 \end{bmatrix}\right) \text{Exp}(\text{x}_2|\lambda)$$

$$f_s(\mathbf{x}) = \mathcal{N}\left((x_0, x_1)|(1,1), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \text{Exp}(\text{x}_2|2),$$

with $\mathcal{N}$ indicating a Normal Gaussian distribution and Exp an exponential distribution. In the likelihood above, b and s determine the mixture of background and signal, while r and λ control the mean of x_0 and x_2 for the background process respectively. The full parameter space is $\Theta = (S \times R \times \Lambda \times B) \subset \mathbb{R}^4$, with $S = [0, 100]$, $R = [-5, 5]$, $\Lambda = [0, 10]$ and $B = [700, 1300]$.

De Castro and Dorigo (2019) use this 3D synthetic mixture example to showcase the benefits of their proposed summary statistics, which are built to minimize the effect of nuisance parameters over the parameters of interest by minimizing a Poisson pseudo-likelihood, against classification-based summary statistics. This example is interesting because of the presence of multiple background-related nuisance parameters which are not of interest (r, λ and b) and a significant overlap between signal and background distributions. Figure 2.10 shows the univariate and bivariate distributions of 50000 samples from the mixture distribution, separated between signal and background. Here we will use the same example to construct confidence sets for the intensity of the signal process s, using the mixture defined in equation (2.15) as the forward simulator F_θ. Note that this setup is more challenging than the one in Section 2.9.3 for the closeness of the background and signal distribution and the presence of nuisance parameters. We consider two different settings:

(I) Estimating s with b, r, λ as nuisance parameters (this corresponds to benchmark 4 in De Castro and Dorigo 2019). Here $\Theta_{(I)} = \Theta \subset \mathbb{R}^4$;

(II) Estimate s and r simultaneously while fixing b and λ (this corresponds to benchmark 1 in De Castro and Dorigo 2019 with no nuisance parameters). Here $\Theta_{(II)} = (S \times R \times \{\lambda_0\} \times \{b_0\}) \subset \mathbb{R}^2$, with $\lambda_0 = 3.0$ and $b = 1000$.

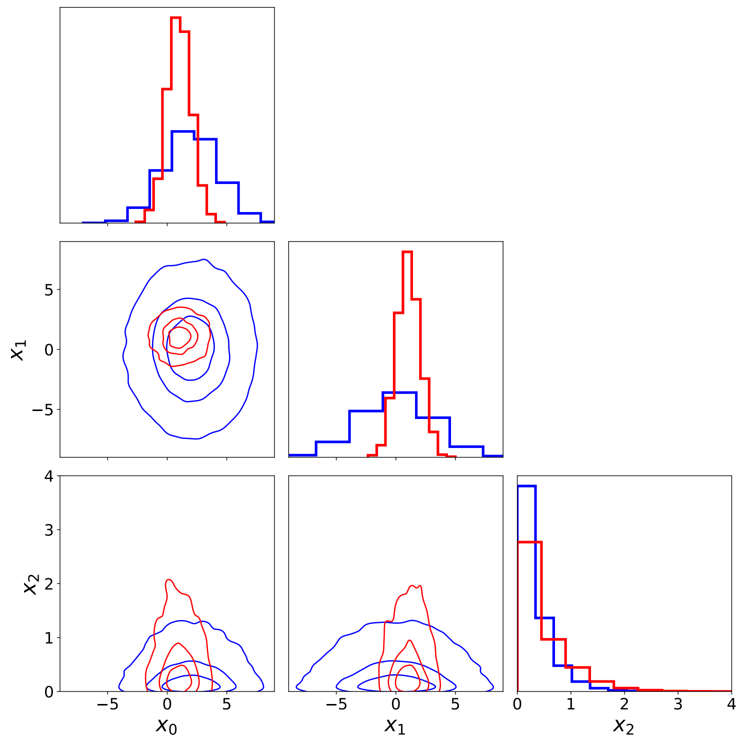


Figure 2.10: Univariate and bivariate distributions of 50000 samples for background (blue) and signal (red) distribution from the 3D mixture in Equation (2.15), with $(s, r, \lambda, b) = (50, 0, 3, 1000)$ (as in De Castro and Dorigo 2019, Figure 2).

For our constructed confidence sets, we consider the set average power, size and coverage as performance metrics. The average power is defined as the proportion of the parameter space outside of the true parameter value which is not covered by the confidence set. In other words, it is a measure of the Type II error in terms of ratios of the parameter space; the highest power would be 1.0 (the confidence set excludes all the values which are not the true parameter), while the lowest would be 0 (the confidence set covers the entirety of the parameter space). Confidence set size is on the other hand here defined as the proportion of parameter space covered by the confidence set (inclusive of the true parameter value). Finally, coverage indicates the proportion of times the true parameter value is included in the confidence set.

For setting (I), we first select the best probabilistic classifier and sample B by evaluating the cross entropy and the odds loss value across different sample sizes B over a held-out simulation set; as shown in Figure 2.11, top panel, the lowest values are achieved by a gradient boosted trees classifier with tree depth equal to 5 and 500 trees at $B = 100000$.

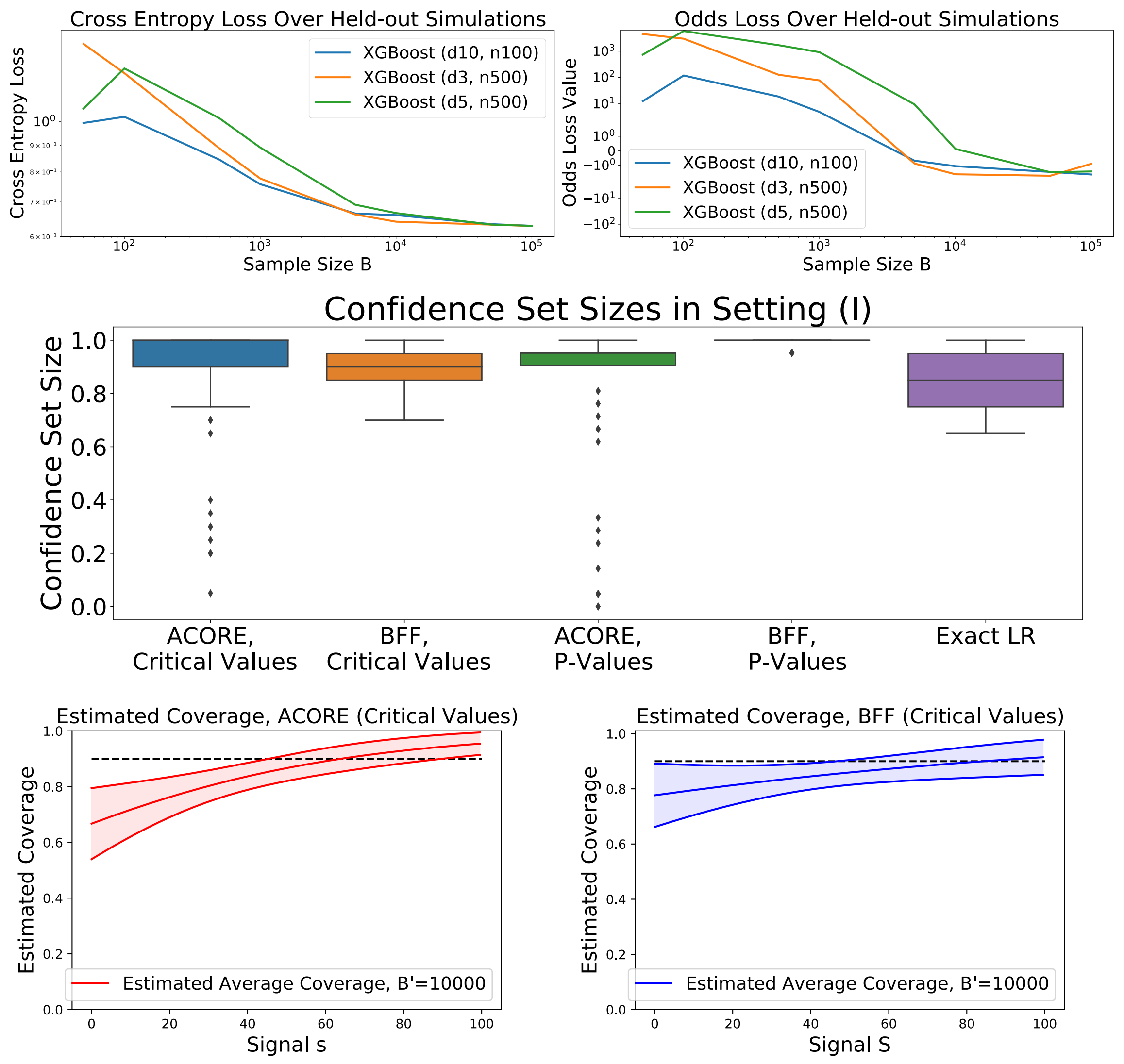


Figure 2.11: Results and diagnostics for setting (I). *Top*: Cross entropy and odds loss over a held-out simulation set for the best performing probabilistic classifiers for odds estimation (see text and Section B.1 for more details). *Center*: Confidence set sizes distribution across 100 repetitions for all methods included in Table 2.4. `BFF` achieves smaller confidence sets more consistently, although not performing as well as the exact likelihood ratio confidence sets. *Bottom*: Estimated confidence sets across the signal parameter space using logistic regression, while keeping the nuisance parameters fixed (see text). `BFF` and `ACORE` confidence sets tend to under-cover for small values of the signal parameter, with the effect being more prominent for `ACORE` .

Table 2.4 reports the confidence sets average power, size and coverage over 100 repetitions with an observed sample size of $n = 25$ (we use $B' = 25000$, deep quantile regression and a budget for maximization/integration of 5000 samples). All methods achieve valid confidence sets, with the coverage within or above the expected variability over 100 repetitions (which is $[0.84, 0.95]$). The confidence sets estimated with the exact likelihood ratio (LR) are included for comparison. As shown in Figure 2.11, center panel, `BFF` using critical value estimation seem to achieve the smallest confidence set sizes among all estimation methods. However, the variance in the size and power of `ACORE` is much more significant. `ACORE` and `BFF` indeed approach nuisance parameters differently by profiling (i.e., maximizing) and integration, respectively, which could explain the difference in variability across the two methods. Overall, the reported confidence sets are wide due to the sample size, currently set to $n = 25$; with this observed sample size, one would observe very few samples from the signal distribution f_s [**]. Figure 2.12 shows how the confidence sets for `ACORE` and `BFF` are smaller the larger the observed sample size n, but at a slow rate due to both the overlap of the background and signal distribution and the few samples extracted in proportion from the signal distribution.

Setting (I) – s parameter of interest, r, λ, b nuisance parameters

Cutoff Type	*Test Statistic*	*Average Power*	*CI Size*	*Coverage*
Crit. Values	`ACORE`	0.102 ± 0.205	0.90 ± 0.20	0.91
	`BFF`	0.121 ± 0.073	0.88 ± 0.07	0.90
p-values	`ACORE`	0.206 ± 0.255	0.84 ± 0.27	0.88
	`BFF`	0.050 ± 0.011	1.00 ± 0.01	1.00
-	Exact LR.	0.139 ± 0.117	0.86 ± 0.12	1.00

Table 2.4: Average power, size and coverage for `ACORE` and `BFF` confidence sets for setting (I), using both critical value and p-value estimation over 100 repetitions with an observed sample size $n = 25$. We use $B = 100000$, $B' = 25000$, a gradient boosted trees classifier for odds estimation, a deep quantile regression for critical value regression and a multilayer perceptron for p-value estimation. Confidence sets constructed with the exact likelihood ratio included for comparison. `BFF` confidence sets with critical values provide the smallest confidence sets consistently, while `ACORE` confidence sets show much more variability (see text).

[**] Given that the true values for signal and background are 50 and 1000 in setting (I), only $50/1050 \approx 4.7\%$ of the observed samples come from the signal distribution f_s. For $n = 25$ samples, this means 1 sample on average.

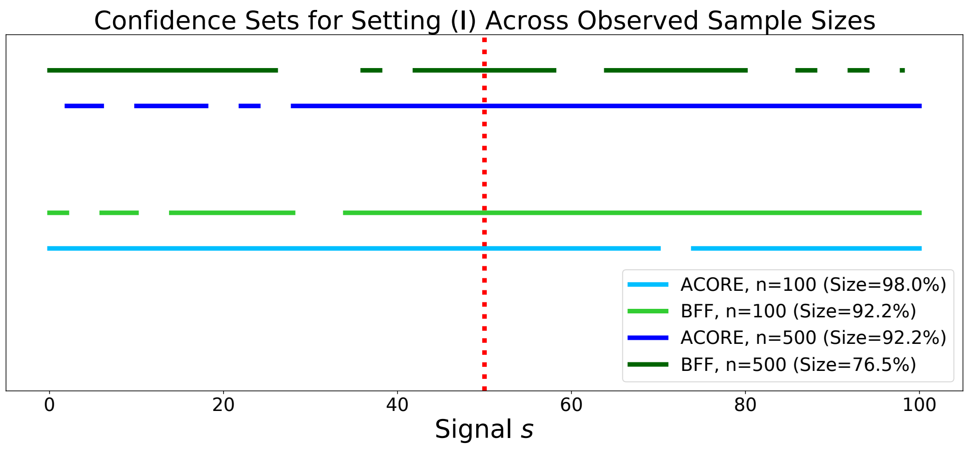


Figure 2.12: ACORE and BFF confidence sets for the signal parameter s in setting (I), with the same setup as Table 2.4, but increasing the observed sample size to $n = 100$ (bottom) and $n = 500$ (top). Both confidence sets decrease in size when increasing the observed sample with respect to the results in Table 2.4 (with BFF more so than ACORE) and include the true parameter (red vertical line), but they remain relatively wide due to the characteristic of the mixture model (see text).

Finally, Figure 2.11 (bottom panels) shows the estimated coverage for different signal values using our diagnostic tools (prediction intervals are obtained via logistic regression). Since in this setting the full parameter space $\Theta_{(I)}$ has dimension 4, for visualization purposes we check our estimate across the one-dimensional signal space when setting the nuisance parameter values at $(r, \lambda, b) = (0, 3, 1000)$ (that is, the same as in Table 2.4). For finite $B' = 25,000$, both ACORE and BFF seems to be providing valid confidence sets only for larger values of the signal, which includes $s = 50$ (the true parameter value for the experiment shown in Table 2.4), with the effect being more prevalent in ACORE .

For setting (II), we follow a similar procedure as in setting (I). Firstly, we select a multilayer perceptron with two hidden layers to learn the odds values with a sample $B = 100000$, as shown in Figure 2.13, top panel. Table 2.5 reports the confidence sets average power, size and coverage over 100 repetitions with an observed sample size of $n = 25$ (we use $B^{'} = 25000$, deep quantile regression and a budget for maximization/integration of 2000 samples), again adding the exact likelihood ratio confidence sets for comparison. As in setting (I), all methods achieve valid confidence sets, with the coverage within or above the expected variability over 100 repetitions ($[0.84, 0.95]$). Confidence sets constructed with both ACORE and BFF via critical value estimation seem to provide the smaller confidence sets among estimated confidence sets, although not on par with the ones built using the exact likelihood ratio, as illustrated in Figure 2.13 (bottom panel). Using p-values estimation to construct confidence sets seems to lead to a degradation in performance, as p-value estimation is negatively affected by the uniform sampling from the simulated set $\mathcal{T}'_{B'}$ over a two dimensional parameter space.[††]

Setting (II) – s, r parameters of interest, no nuisance parameters

Cutoff Type	*Test Statistic*	*Average Power*	*CI Size*	*Coverage*
Crit. Values	ACORE	0.205 ± 0.192	0.80 ± 0.19	0.92
	BFF	0.191 ± 0.107	0.81 ± 0.11	0.88
p-values	ACORE	0.205 ± 0.237	0.80 ± 0.24	0.86
	BFF	0.091 ± 0.101	0.92 ± 0.10	0.99
	Exact LR.	0.247 ± 0.155	0.75 ± 0.15	0.90

Table 2.5: Average power, size and coverage for ACORE and BFF confidence sets for setting (II), using both critical value and p-value estimation over 100 repetitions with an observed sample size $n = 25$. We use $B = 100000$, $B' = 25000$, a two-layer multilayer perceptron for odds estimation, a deep quantile regression for critical value regression and a multi-layer perceptron for p-value estimation. Confidence sets constructed with the exact likelihood ratio included for comparison. BFF and ACORE confidence sets with critical values are the most powerful, although not on par with exact likelihood ratio confidence sets (see text).

Figure 2.14 shows the upper limit of the prediction interval for estimated coverage across the two-dimensional parameter space $\Theta_{(II)}$ for ACORE (top panel) and BFF (bottom panel). The coverage is satisfactory, with the vast majority of the upper limit of the prediction intervals being correctly above the nominal confidence level (here $1 - \alpha = 90\%$). It is also interesting to note the symmetry in coverage around $r = -1$, which is the value for which the signal and background distribution achieve the same mean for the first dimension (see equation 2.15).

[††]See Section 5.2 for a potential solution to improve sampling efficiency for both critical and p-value estimation.

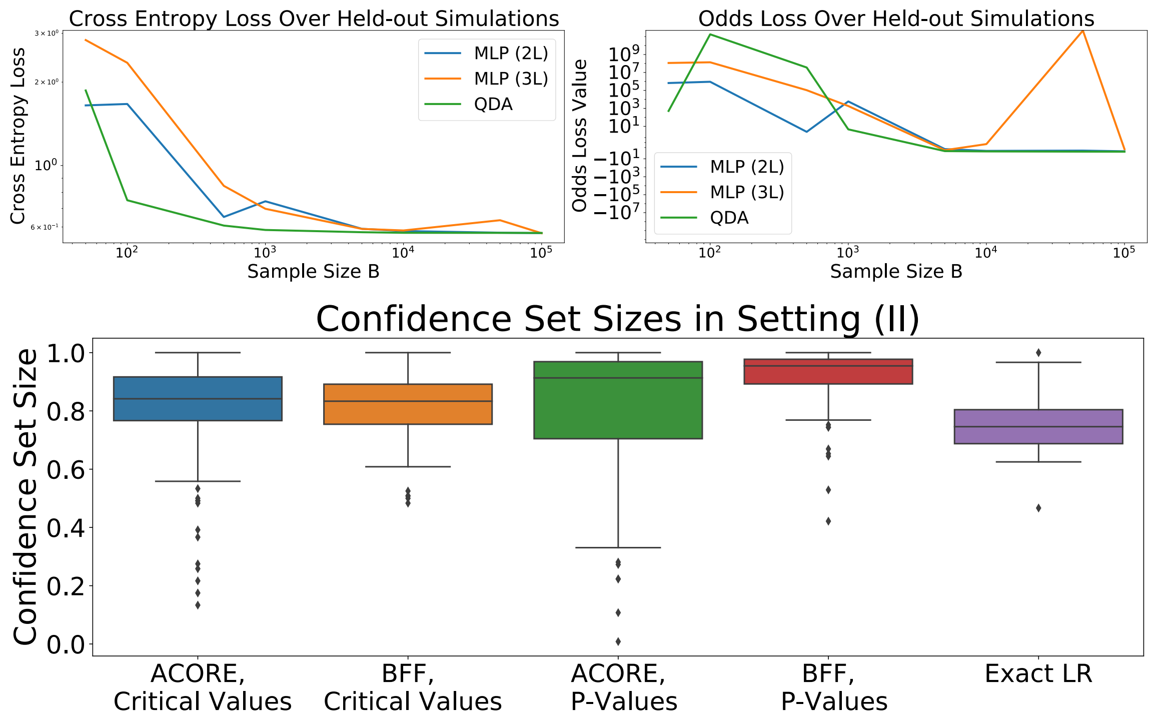


Figure 2.13: *Top*: Cross entropy and odds loss over a held-out simulation set for the best performing probabilistic classifiers for odds estimation in setting (II) (see text and Section B.1 for more details). *Bottom*: Confidence set sizes distribution across 100 repetitions for all methods included in Table 2.4. `ACORE` and `BFF` confidence sets achieve smaller confidence sets when using critical value for confidence sets as opposed to p-values (see text for more details).

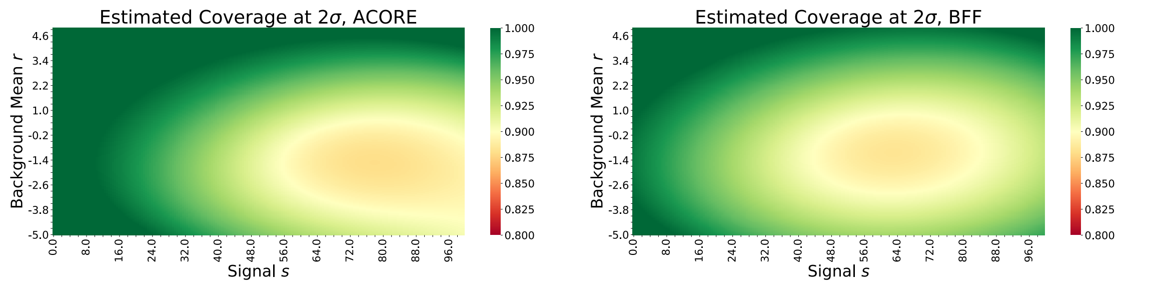


Figure 2.14: Upper limit (mean plus two standard deviations) of the predicted confidence interval across the two dimensional parameter space $\Theta_{(II)}$ for setting (II) for `ACORE` (top) and `BFF` (bottom) confidence sets. The nominal coverage level (90%) is correct across the vast majority the parameter space for both `ACORE` and `BFF` (that is, below the upper limit of the estimated prediction interval).

2.9.5 Galaxy Images Inference

In this section we provide an example of constructing confidence intervals in a setting with high-dimensional simulated image data. We use the open-source galaxy simulator GALSIM (Rowe et al., 2015), which allows to generated realistic images of astronomical objects and integrates real observational effects such as pixelization and blurring. In our simulation we consider globular galaxies, that is a spherical group of stars rotating around the galaxy center, following the same setup as in Izbicki et al. (2014). We consider two parameters $\theta = (\alpha, \lambda)$: the orientation with respect to the x-axis of the image α and the axis ratio of the galaxy λ, resulting in the parameter space $\Theta = [-\pi, \pi] \times [0, 1] \subset \mathbb{R}^2$. We down-sample the simulated galaxy images to a 20×20 resolution to mimic realistic observations.

In our experiment we construct a confidence set based on a single 400-dimensional galaxy image ($n = 1$). We set the true parameter to be $\theta^\star = (0, 0.5)$. Table 2.6 reports the ACORE and BFF confidence sets average power, size and coverage of confidence sets constructed with four different probabilistic classifiers for odds estimation — MLP, an adapted version of Alexnet (Krizhevsky et al., 2012) and two residual networks (He et al., 2016) — as well the cross-entropy and odds loss calculated over a separate validation set. Results are reported over 25 repetitions, with $B = 100000$, $B^{'} = 250000$, $M = 2500$, with all constructed ACORE and BFF confidence sets being valid. Confidence sets average power and size are computed in the same way as in Section 2.9.4. As illustrated by Table 2.6, the adapted Alexnet architecture, which achieves the smallest cross entropy and odds loss, constructs confidence sets with the higher power and smaller set size for both ACORE and BFF. This remarks the importance of our practical strategy for choosing probabilistic classifiers, as the higher capacity ResNet architecture actually overfit the training set, hence achieving a worse cross entropy and odds loss on the validation set. When comparing ACORE and BFF using the adapted Alexnet architecture, we note that the size of the BFF confidence sets is significantly lower, as also shown in Figure 2.15. In this setup there are indeed fewer sources of error for BFF than ACORE, as the denominator in the BFF test statistic with a sample size $n = 1$ is identically one, which here translates in confidence sets with a higher power and smaller size.

Galsim 2D inference on $\theta = (\alpha, \lambda)$ with $n = 1$

Test Statistic	*Classifier*	*Average Power*	*CI Size*	*Coverage*	*CE Loss*	*Odds Loss*
ACORE	Alexnet	0.171 ± 0.175	0.83 ± 0.17	1.00	0.6931	-1.0008
	MLP	0.066 ± 0.014	0.93 ± 0.01	1.00	0.6933	-0.9985
	Resnet34	0.134 ± 0.143	0.87 ± 0.14	1.00	0.6990	-0.9885
	Resnet18	0.033 ± 0.077	0.97 ± 0.08	1.00	0.6985	-0.9762
BFF	Alexnet	0.337 ± 0.029	0.66 ± 0.03	1.00	0.6931	-1.0008
	MLP	0.068 ± 0.024	0.93 ± 0.02	1.00	0.6933	-0.9985
	Resnet34	0.009 ± 0.006	0.99 ± 0.01	1.00	0.6990	-0.9885
	Resnet18	0.000 ± 0.001	1.00 ± 0.00	1.00	0.6985	-0.9762

Table 2.6: Average power, size and coverage for ACORE and BFF confidence sets for the simulated galaxy image example, using critical value estimation and an observed sample size of $n = 1$. Also included the cross-entropy and odds loss on a separate validation set of simulations for the four classifiers used to learn the odds. We use $B = 100000$, $B^{'} = 25000, M = 2500$ and a deep quantile regression for critical value regression. The best classifier (Alexnet) achieves the highest power and smallest size for BFF confidence sets, for which the integration step is not needed since $n = 1$, as opposed to the ACORE test statistics in which calculating the maximizer over the parameter space is still required (see text).

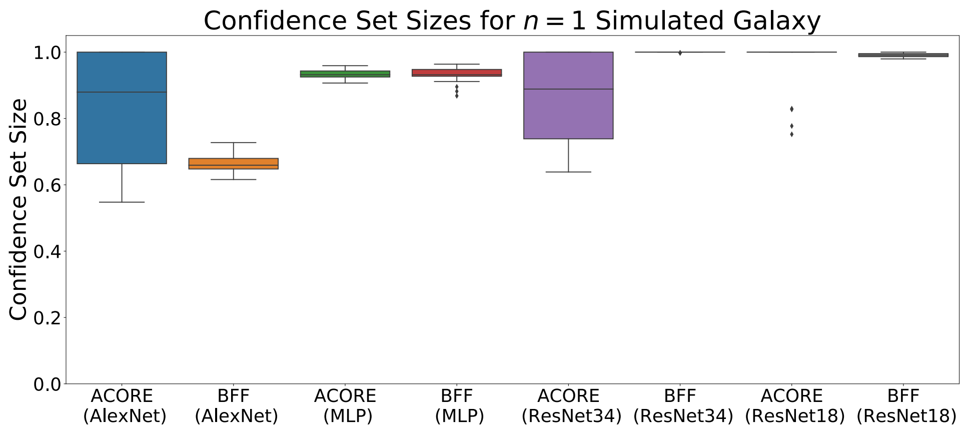


Figure 2.15: Confidence set size distribution given $n = 1$ 400-dimensional simulated galaxy image, in the setting described by Table 2.6. BFF confidence set require estimating fewer quantities than ACORE when the observed sample size is $n = 1$, resulting in smaller and more powerful confidence sets (see text).

2.10 Comparison with Existing Methodologies

In this section we compare our proposed test statistics with two frequentist likelihood-free inference approaches: Gaussian process interpolation and calibrated approximate ratio of likelihood. We analyse the performance of both methods on the one-dimensional Poisson and GMM example from section 2.9.1 and comparing them against the performance of the `ACORE` test statistic. Given the very similar performance achieved by `ACORE` and `BFF` on such toy examples (see Tables 2.9.1 and 2.9.1), as well as their identical computational runtime in the current implementation (see Section 2.12), we only provide a comparison for `ACORE`.

2.10.1 Monte Carlo Synthetic Likelihood-Based Methods

In this section we compare the performance of `ACORE` with Monte-Carlo (MC) synthetic likelihood-based methods, more specifically Gaussian process (GP) interpolation (Frate et al., 2017). The latter method first simulates multiple sample points for a few different values of θ. For each fixed θ, one fits a Gaussian synthetic likelihood function. The GP likelihood model is then used to smoothly interpolate across the parameter space by fitting a mean function $m(\theta)$ and a covariance function $\Sigma(\theta)$. As a note, Cranmer et al. (2020) point out that such MC methods are less efficient than methods that estimate the likelihood ratio directly because of the need to first estimate the entire likelihood.

For our comparison, we use the two toy examples described in Section 2.9.1 and Table 2.1. To allocate B sample points for the GP interpolation, we use the following strategy: For $q \in \{5, 10, 25\}$, first choose $\theta_1, ..., \theta_q$ on an evenly spaced grid across the parameter space. Then, generate $N = B/q$ sample points $\mathbf{X}_1, ..., \mathbf{X}_N$ at each location θ.

Tables 2.7 and 2.8 summarize the results. Unlike Section 2.9.1, we do not report the cross-entropy loss because GP interpolation is not a classification algorithm; instead we report the mean squared error in estimating the likelihood ratio across the parameter space. Our results show that when the simulated data at each θ are approximately Gaussian, as in the Poisson example, MC-based GP interpolation provides a better approximation of the likelihood ratio due to its parametric assumptions. However, when the parametric assumptions are not valid, as in the GMM example, MC-based GP fails to approximate the likelihood ratio regardless of how large N or B are. In such settings, we do better with a fully nonparametric approach. As a note, MC-based GP uses the asymptotic χ^2 approximation by Wilks' theorem to determine the critical values of the confidence sets. In our experiments, using quantile regression for critical values instead led to a significant increase in power for the GP likelihood models: from ≈ 0.48 to ≈ 0.51 for the Poisson example, and from ≈ 0.02 to ≈ 0.2 for the GMM example. All fitted classifiers produce valid 90% confidence sets for θ according to our diagnostics.

Poisson Example

B	*Classifier*	*90 % Mean Squared Error Interval*	*Average Power*	*Size of Confidence Set [%]*
100	MLP	[2.14, 989.78]	0.27	72.8 ± 16.4
	NN	[4.14, 4074.65]	0.25	75.6 ± 23.2
	QDA	[0.41, 34.79]	0.41	**60.1 ± 14.9**
	G.P. (5)	**[0.05, 4.09]**	0.47	**53.5 ± 9.2**
	G.P. (10)	**[0.06, 4.97]**	**0.48**	**53.2 ± 10.7**
	G.P. (25)	**[0.03, 6.54]**	**0.48**	**53.2 ± 10.8**
500	MLP	[0.86, 22.45]	0.38	62.2 ± 19.1
	NN	[1.95, 32.78]	0.37	**64.2 ± 17.3**
	QDA	[0.08, 6.95]	0.45	**55.5 ± 10.8**
	G.P. (5)	**[0.01, 0.81]**	**0.49**	**52.4 ± 5.6**
	G.P. (10)	**[0.02, 0.85]**	**0.49**	**52.0 ± 5.4**
	G.P. (25)	**[0.01, 1.12]**	**0.48**	**52.5 ± 6.0**
1,000	MLP	[0.81, 21.44]	0.42	**58.8 ± 17.0**
	NN	[1.77, 17.88]	0.45	**56.1 ± 16.2**
	QDA	[0.06, 2.83]	**0.49**	**52.1 ± 9.0**
	G.P. (5)	**[0.01, 0.48]**	**0.49**	**52.3 ± 5.0**
	G.P. (10)	**[0.01, 0.46]**	**0.48**	**52.5 ± 5.3**
	G.P. (25)	**[0.01, 0.45]**	**0.48**	**52.6 ± 5.5**
-	**Exact**	-	**0.54**	**45.0 ± 4.9**

Table 2.7: Results for `ACORE` (MLP, NN, QDA) and Gaussian Process interpolation (GP for $q = 5, 10, 25$; see text) for the Poisson example of Section 2.9.1. The table lists the 5th and 95th quantiles of the mean squared error (MSE) between the estimated and true likelihood, the power (averaged over θ) and the average and one standard deviation of the size of confidence sets, for different values of B and for different classifiers. Best results for each training sample size B are marked in bold-faced, which all classifiers achieve with $B = 1000$.

GMM Example

B	*Classifier*	*90 % Mean Squared Error Interval* ($\times 10^3$)	*Average Power*	*Size of Confidence Set [%]*
100	MLP	**[0.34, 1.46]**	**0.87**	**14.5 ± 4.5**
	NN	[1.33, 11.77]	0.49	52.1 ± 24.7
	QDA	[2.88, 3.56]	0.16	84.0 ± 21.8
	G.P. (5)	[3.35, 3.82]	0.02	97.7 ± 8.8
	G.P. (10)	[3.34, 3.82]	0.03	96.9 ± 9.5
	G.P. (25)	[3.36, 3.82]	0.02	98.2 ± 6.1
500	MLP	**[0.44, 1.35]**	**0.90**	**12.1 ± 2.8**
	NN	[0.99, 2.65]	0.57	44.0 ± 23.3
	QDA	[3.14, 3.73]	0.16	83.8 ± 22.2
	G.P. (5)	[3.39, 3.83]	0.00	100.0 ± 0.0
	G.P. (10)	[3.39, 3.83]	0.01	99.1 ± 5.5
	G.P. (25)	[3.38, 3.83]	0.00	99.8 ± 1.5
1000	MLP	**[0.53, 1.17]**	**0.90**	**12.1 ± 2.8**
	NN	[0.57, 2.04]	0.71	30.2 ± 18.5
	QDA	[3.26, 3.94]	0.14	85.7 ± 20.1
	G.P. (5)	[3.39, 3.98]	0.00	100.0 ± 0.0
	G.P. (10)	[3.39, 3.98]	0.00	100.0 ± 0.0
	G.P. (25)	[3.39, 3.98]	0.00	99.9 ± 1.2
-	**Exact**	-	**0.92**	**9.5 ± 2.0**

Table 2.8: Results for ACORE (MLP, NN, QDA) and Gaussian Process interpolation (GP for $q = 5, 10, 25$; see text) for the GMM example of Section 2.9.1. The table lists the 5th and 95th quantiles of the mean squared error (MSE) between the estimated and true likelihood, the power (averaged over θ) and the average and one standard deviation of the size of confidence sets, for different values of B and for different classifiers. Best results for each training sample size B are marked in bold-faced, which are achieved by ACORE using an MLP classifier with $B = 1000$.

2.10.2 Calibrated Approximate Ratio of Likelihood Classifiers

In this section we compare the performance of ACORE with the calibrated approximate ratio of likelihood (CARL) estimator by Cranmer et al. (2015). CARL approximates the likelihood ratio $\Lambda(D; \Theta_0) = \mathcal{L}(D; \theta_0)/\mathcal{L}(D; \theta_1)$ by turning the density ratio estimation into a supervised classification problem, where a probabilistic classifier is trained to separate samples from F_{θ_0} and F_{θ_1}. As such, CARL classifiers are "doubly parameterized" by θ_0 and θ_1, whereas the ACORE classifier is parameterized by a *single* parameter θ in the definition of the odds of F_θ versus G.

In our study, we include three different CARL classifiers, implemented with the `MADMINER` neural network-based software (Brehmer et al., 2020a): (a) a shallow perceptron with 100 neurons (equivalent to the MLP used in Section 2.9.1), (b) a 2-layer deep network with 20 neurons per layer, and (c) a 2-layer deep network with 20 and 50 neurons in the two layers respectively.[‡‡] To allocate B sample points for interpolation we devised two schemes: (i) a uniform sampling, and (ii) a Monte Carlo sampling over the parameter space. For (i), we uniformly sample B parameters and then generate a sample point $\mathbf{X}$ at each parameter value. For (ii), we first select evenly spaced parameters $\theta_{0,1}, ..., \theta_{0,q}$ and $\theta_{1,1} ..., \theta_{1,q}$, for the numerator and the denominator respectively. We set $q \in \{10, 20, 30\}$, resulting in $N = B/q$ sample points $\mathbf{X}_1, ..., \mathbf{X}_N$ at each θ location. Because the χ^2 approximation by Wilks' theorem did not yield valid confidence sets for CARL classifiers, we computed critical values as in Algorithm 4. Tables 2.9 and 2.10 show the results of `ACORE` and `CARL` for the synthetic data in Section 2.9.1 and Table 2.1. For both the Poisson and GMM examples, CARL classifiers yield a higher mean squared error in estimating the likelihood ratio, as well as lower power and larger confidence intervals.

As a final proof of concept, we analyse the statistical loss in efficiency when learning the odds ratio by using a doubly parametrized decision function $s(\mathbf{X}; \theta_1, \theta_2)$ rather than a single parametrized decision function. Let $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ be the feature space of dimension d, and $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$ be the parameter space of dimension d_θ. Let $\mathcal{Z}_1 = (\mathcal{X} \times \Theta) \subset \mathbb{R}^{d+d_\theta}$ and $\mathcal{Z}_2 = (\mathcal{X} \times \Theta \times \Theta) \subset \mathbb{R}^{d+2d_\theta}$. For simplicity, we use a nearest neighbor classifier to learn the odds (Section 2.2). Assume the probabilities estimated in Algorithm 1 satisfy the weak margin assumption Döring et al. (2017), so that for every $0 < t \leq 1$

$$\mathbb{P}(0 < \mathbb{P}(Y = 1 | X, \theta) \leq t) \leq c^\star \cdot t^\alpha,$$

for some $c^\star > 0$ and $\alpha > 0$. (We assume the same α holds for both `ACORE` and `CARL`.) Generate an ϵ-ball covering of the space $\mathcal{Z}_1$, so that $S_1 \subseteq \bigcup_i B_\epsilon(\mathbf{x}_i, \theta_i)$; equivalently, let S_2 be the ϵ-ball covering of Z_2, so $S_2 \subseteq \bigcup_i B_\epsilon(\mathbf{x}_i, \theta_{0,i}, \theta_{1,i})$. The covering number is of the following order:

[‡‡]Changing the number of neurons per layers did not seem to provide a significant difference in performance for the 2-layer deep networks. Number of epochs and learning rate were manually tuned (with a search in the range $[20, 200]$ and $10^{\{-6,\ldots,-2\}}$ respectively).

$$|S_1| = \mathcal{O}\left(\epsilon^{-d-d_\theta}\right) \quad , \quad |S_2| = \mathcal{O}\left(\epsilon^{-d-2d_\theta}\right)$$

Finally, assume that at any $(\mathbf{x}, \theta) \in \mathcal{Z}_1$, the probability $P(Y=1|\mathbf{x}, \theta)$ can be evaluated by smooth interpolation without error degradation if another point is placed within 2ϵ, for some $\epsilon > 0$ (same for $\mathcal{Z}_2$).

Lemma 2.2.1. *Consider the setup above. Assume $N = \mathcal{O}\left(n\epsilon^{-d-d\theta}\right)$ total samples, with samples divided uniformly in each ball. The relative efficiency between* `ACORE` *and* `CARL` *is of the following order:*

$$\mathcal{O}\left(n^{-\frac{d_\theta(\alpha+1)}{(d+d_\theta)(d+2d_\theta)}} \cdot \epsilon^{\frac{d_\theta(\alpha+1)}{d+2d_\theta}}\right).$$

The first term degrades the efficiency due to nearest neighbor convergence in $\mathcal{Z}_2$ rather than $\mathcal{Z}_1$, while the second term for the larger number of ϵ-ball in $\mathcal{Z}_2$. If $d >> d_\theta$, the relative efficiency is of the order:

$$\mathcal{O}\left(n^{-1/d^2} \cdot \epsilon^{1/d}\right) \tag{2.16}$$

Proof. If samples are divided uniformly in each ball, then in $\mathcal{Z}_1$ each ball centroid is estimated with $\mathcal{O}(n)$ samples. Equivalently, in $\mathcal{Z}_2$ each ball centroid is estimated with $\mathcal{O}\left(n\epsilon^{d_\theta}\right)$ samples. The result is obtained by using the convergence rate for nearest neighbor in (Döring et al., 2017, Theorem 1) and taking the ratio between the convergence rate of `ACORE` over the convergence rate of `CARL`. □

Poisson Example

B	*Classifier*	*90 % Mean Squared Error Interval*	*Average Power*	*Size of Confidence Set [%]*
200	MLP	[3.25, 1305.45]	0.17	82.7 ± 15.0
	NN	[2.88, 185.47]	0.34	66.9 ± 20.7
	QDA	**[0.20, 25.16]**	**0.45**	**55.8 ± 13.2**
	MLP (MC)	[2.51, 38.10]	0.24	76.1 ± 21.3
	(20,20) DNN (MC)	[2.53, 25.41]	0.19	80.9 ± 17.8
	(50,20) DNN (MC)	[2.76, 26.00]	0.19	81.3 ± 17.8
	MLP (U)	[2.03, 45.19]	0.19	81.3 ± 19.2
	(20,20) DNN (U)	[2.95, 19.76]	0.24	76.6 ± 19.8
	(50,20) DNN (U)	[2.43, 18.72]	0.23	77.8 ± 20.1
800	MLP	[1.69, 450.81]	0.27	73.0 ± 20.1
	NN	[1.47, 19.32]	0.42	**59.2 ± 15.9**
	QDA	**[0.04, 5.03]**	**0.49**	**52.0 ± 9.3**
	MLP (MC)	[2.38, 24.50]	0.22	78.5 ± 21.0
	(20,20) DNN (MC)	[2.49, 21.49]	0.25	75.3 ± 18.8
	(50,20) DNN (MC)	[2.52, 18.13]	0.23	76.9 ± 20.1
	MLP (U)	[2.04, 23.24]	0.20	79.9 ± 17.4
	(20,20) DNN (U)	[2.48, 17.36]	0.22	77.9 ± 17.6
	(50,20) DNN (U)	[2.25, 17.87]	0.21	78.9 ± 20.0
1,800	MLP	[0.81, 19.11]	0.37	**63.7 ± 21.1**
	NN	[1.09, 11.27]	0.44	**56.9 ± 14.3**
	QDA	**[0.03, 1.60]**	**0.50**	**51.0 ± 6.6**
	MLP (MC)	[2.13, 35.39]	0.18	82.4 ± 17.7
	(20,20) DNN (MC)	[2.74, 28.15]	0.20	80.3 ± 19.7
	(50,20) DNN (MC)	[2.62, 28.15]	0.18	81.9 ± 19.5
	MLP (U)	[2.15, 25.51]	0.19	81.4 ± 19.8
	(20,20) DNN (U)	[2.34, 15.93]	0.23	77.0 ± 22.6
	(50,20) DNN (U)	[2.38, 17.97]	0.19	81.6 ± 17.2
-	**Exact**	-	**0.54**	**45.0 ± 4.9**

Table 2.9: Results for ACORE (MLP, NN, QDA) and CARL or uniform (U) and Monte-Carlo (MC) sampling schemes in the Poisson example of settings of Section 2.9.1. The table lists the 5th and 95th quantiles of the mean squared error (MSE) between the estimated and true likelihood, the power (averaged over θ) and the average and one standard deviation of the size of confidence sets, for different values of B and for different classifiers. The best results for each training sample size B are marked in bold-faced, which is achieved by all classifiers employed by ACORE with $B = 1000$ in terms of confidence size, and by ACORE with QDA in terms of MSE.

GMM Example				
B	*Classifier*	*90 % Mean Squared Error Interval* ($\times 10^3$)	*Average Power*	*Size of Confidence Set [%]*
200	MLP	**[0.56, 1.69]**	**0.88**	**14.2** $\pm$ **8.2**
	NN	[1.13, 4.17]	0.50	51.5 $\pm$ 24.8
	QDA	[3.05, 3.63]	0.12	87.6 $\pm$ 19.7
	MLP (MC)	[3.03, 3.61]	0.27	73.5 $\pm$ 20.5
	(20,20) DNN (MC)	[3.13, 3.70]	0.25	75.6 $\pm$ 20.0
	(50,20) DNN (MC)	[3.16, 3.67]	0.28	72.8 $\pm$ 19.6
	MLP (U)	[3.01, 3.72]	0.30	70.2 $\pm$ 21.2
	(20,20) DNN (U)	[3.18, 3.87]	0.24	76.3 $\pm$ 21.5
	(50,20) DNN (U)	[3.12, 3.92]	0.27	73.0 $\pm$ 21.2
800	MLP	**[0.89, 1.59]**	**0.90**	**12.1** $\pm$ **2.5**
	NN	[0.78, 2.31]	0.69	32.0 $\pm$ 18.9
	QDA	[3.23, 3.66]	0.14	86.1 $\pm$ 20.4
	MLP (MC)	[3.02, 3.58]	0.30	70.8 $\pm$ 20.4
	(20,20) DNN (MC)	[3.10, 3.63]	0.27	73.6 $\pm$ 20.2
	(50,20) DNN (MC)	[3.03, 3.47]	0.30	70.5 $\pm$ 18.5
	MLP (U)	[3.01, 3.62]	0.26	74.7 $\pm$ 20.6
	(20,20) DNN (U)	[3.12, 3.64]	0.26	74.4 $\pm$ 19.2
	(50,20) DNN (U)	[3.00, 3.56]	0.29	71.8 $\pm$ 19.9
1,800	MLP	**[0.33, 1.55]**	**0.90**	**11.5** $\pm$ **2.6**
	NN	**[0.32, 1.57]**	0.83	**19.3** $\pm$ **10.3**
	QDA	[3.29, 3.81]	0.16	83.7 $\pm$ 22.2
	MLP (MC)	[2.99, 3.54]	0.33	67.5 $\pm$ 19.6
	(20,20) DNN (MC)	[3.02, 3.54]	0.31	69.7 $\pm$ 19.3
	(50,20) DNN (MC)	[2.95, 3.51]	0.38	63.1 $\pm$ 15.9
	MLP (U)	[2.99, 3.45]	0.33	67.7 $\pm$ 17.0
	(20,20) DNN (U)	[3.02, 3.56]	0.33	67.3 $\pm$ 18.0
	(50,20) DNN (U)	[2.98, 3.41]	0.38	63.1 $\pm$ 15.3
-	**Exact**	-	**0.92**	**9.5** $\pm$ **2.0**

Table 2.10: Results for `ACORE` (MLP, NN, QDA) and CARL or uniform (U) and Monte-Carlo (MC) sampling schemes in the GMM example of settings of Section 2.9.1. The table lists the 5th and 95th quantiles of the mean squared error (MSE) between the estimated and true likelihood, the power (averaged over θ) and the average and one standard deviation of the size of confidence sets, for different values of B and for different classifiers. The best results for each training sample size B are marked in bold-faced, which are achieved by `ACORE` using an MLP classifier with $B = 1000$.

2.11 Runtime Analysis

In this section we provide a runtime analysis for constructing one ACORE confidence set for the two examples in Section 2.9.1. We also provide a running time comparison with the two methods described in Sections 2.10.1 and 2.10.2. This analysis was performed on a 8-Core Intel Xeon 3.33GHz X5680 CPU. The procedure for constructing confidence sets with ACORE and BFF is outlined in Algorithm 3. In this analysis we break the computation into 4 steps: (i) odds training as described by Algorithm 1, (ii) computing the test statistic for the observed data, (iii) computing the test statistic in the B' sample as described by Algorithm 4 and (iv) quantile regression algorithm training. Table 2.11 summarizes our running times results. ACORE constructs one confidence set in less than 20 and 30 seconds for Poisson and GMM examples respectively. The main computational bottleneck is step (iii), while the computation time of step (i) increases with the sample size B. Figure 2.16 shows the results of comparing confidence set construction runtimes with MC GP and CARL classifiers, which highlights how ACORE classifiers are comparable with GP interpolation in terms of running times, while CARL classifiers tend to have significantly longer runtimes.

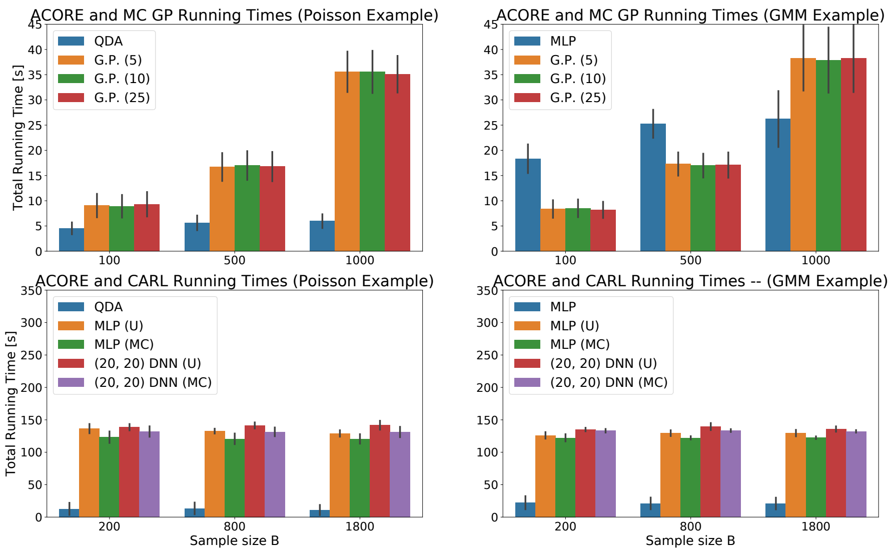


Figure 2.16: Runtimes in seconds for constructing a confidence set for the Poisson example (left panels) and GMM example (right panels). The best ACORE classifier runtime is compared with Gaussian process interpolation (GP) for $q = \{5, 10, 25\}$, and the two smaller CARL classifiers for both sampling schemes. See text for details. Confidence bars are built with a one standard deviation interval around the mean.

Running Times to Generate a Confidence Set (Seconds) – Poisson Example

B	*Classifier*	*Odds Training*	*Odds Prediction*	*Calculate $B^{'}$ Test Statistics*	*Quantile Regression Training*	*Total Running Time*
100	MLP	0.38 ± 0.31	0.42 ± 0.10	10.40 ± 0.71	0.66 ± 0.28	11.86 ± 1.02
	NN	0.03 ± 0.01	0.35 ± 0.12	9.83 ± 4.99	0.82 ± 0.67	11.02 ± 5.73
	QDA	0.02 ± 0.01	0.18 ± 0.11	4.50 ± 2.65	0.58 ± 0.21	5.29 ± 2.96
500	MLP	1.62 ± 0.39	0.46 ± 0.04	11.49 ± 0.45	0.68 ± 0.09	14.26 ± 0.61
	NN	0.13 ± 0.01	0.54 ± 0.03	13.28 ± 0.26	0.66 ± 0.04	14.60 ± 0.29
	QDA	0.13 ± 0.01	0.16 ± 0.01	4.12 ± 0.09	0.65 ± 0.06	5.05 ± 0.14
1,000	MLP	2.65 ± 0.88	0.48 ± 0.08	11.93 ± 1.93	0.73 ± 0.06	15.79 ± 2.30
	NN	0.24 ± 0.04	0.77 ± 0.21	17.90 ± 2.82	0.67 ± 0.10	19.59 ± 2.83
	QDA	0.27 ± 0.08	0.17 ± 0.05	4.37 ± 1.02	0.64 ± 0.16	5.45 ± 1.29

Running Times to Generate a Confidence Set (Seconds) – GMM Example

B	*Classifier*	*Odds Training*	*Odds Prediction*	*Calculate $B^{'}$ Test Statistics*	*Quantile Regression Training*	*Total Running Time*
100	MLP	5.89 ± 1.66	0.45 ± 0.18	10.79 ± 2.06	0.60 ± 0.21	17.74 ± 3.92
	NN	0.03 ± 0.00	0.29 ± 0.06	8.60 ± 2.84	0.61 ± 0.18	9.53 ± 3.05
	QDA	0.03 ± 0.01	0.14 ± 0.04	3.81 ± 1.38	0.52 ± 0.14	4.50 ± 1.57
500	MLP	9.89 ± 1.34	0.43 ± 0.06	11.64 ± 0.64	0.69 ± 0.06	22.64 ± 1.83
	NN	0.17 ± 0.01	0.52 ± 0.04	13.11 ± 0.79	0.63 ± 0.07	14.43 ± 0.85
	QDA	0.16 ± 0.01	0.15 ± 0.02	4.05 ± 0.26	0.59 ± 0.08	4.94 ± 0.35
1,000	MLP	13.40 ± 2.60	0.47 ± 0.09	11.76 ± 0.79	0.68 ± 0.11	26.31 ± 3.36
	NN	0.34 ± 0.09	0.70 ± 0.11	17.15 ± 1.90	0.71 ± 0.17	18.90 ± 2.06
	QDA	0.32 ± 0.05	0.17 ± 0.05	4.75 ± 1.26	0.62 ± 0.07	5.87 ± 1.36

Table 2.11: Runtimes in seconds for constructing a confidence set with `ACORE` for the Poisson example (top) and GMM example (bottom). The procedure for constructing confidence sets is outlined in Algorithm 3, and is split in 4 steps (see text). The rightmost column shows total runtimes.

2.12 Computational Shortcuts

This section details some of expedients necessary to the computation stability of the `ACORE` and `BFF` test statistics. The main points covered are computational complexity and stability.

In the current implementation, the maximization task in `ACORE` (2.5) is computed via grid-search, while the integration in `BFF` (2.6) is obtained via Monte Carlo approximation. Assuming the two operations use the same number of samples M, the complexity of both algorithms in their current implementation is $\mathcal{O}(n_{\text{grid}} \times B^{'} \times M)$, where n_{grid} is the number of parameter values evaluated in the Neyman inversion procedure to construct the confidence set and $B^{'}$ is the number of simulated samples used for critical values or p-values estimation. To avoid a costly `for` loop, computing the odds for the test statistics is done for all the sampled parameter values and simulated data at the same time by creating a matrix with $\mathcal{O}(n_{\text{grid}} \times B^{'} \times M)$ of rows.

In addition, for both ACORE and BFF the product of the odds of the observed data $\prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i^{\text{obs}}; \theta)$ can quickly run into overflow/underflow. Dropping the observed notation for simplicity, if one assumes $m \leq \mathbb{O}(\mathbf{X}_i; \theta_j) \leq M$ for all X_i, θ_j, the product over n samples can range from $m^n \leq \prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta_j) \leq M^n$ which could be below or above machine precision depending on the values of m and M respectively. A solution to this issues is running computations in log-space, which provides computationally stable calculations even for large samples. For ACORE it is enough to note that the product of odds can be written as the exponential of sum of odds, that is:

$$\prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta_0) = \exp\left(\sum_{i=1}^{n} \log(\mathbb{O}(\mathbf{X}_i; \theta_0))\right)$$

The computations for BFF are more involved. By using the Monte Carlo approximation, we obtain that:

$$\begin{aligned}\tau(\mathcal{D}; \theta_0) = \frac{\prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta_0)}{\int_\Theta \left(\prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta)\right) d\pi(\theta)} &\approx \frac{\prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta_0)}{\frac{1}{m}\sum_{j=1}^{m} \prod_{i=1}^{n} \mathbb{O}(\mathbf{X}_i; \theta_j)} \\ &= \frac{\exp^{\sum_{i=1}^{n} \log(\mathbb{O}(\mathbf{X}_i; \theta_0))}}{\frac{1}{m}\sum_{j=1}^{m} \exp^{\sum_{i=1}^{n} \log(\mathbb{O}(\mathbf{X}_i; \theta_j))}}.\end{aligned}$$

Let $\psi^0 = \sum_{i=1}^{n} \log(\mathbb{O}(\mathbf{X}_i; \theta_0))$ and $\psi_j = \sum_{i=1}^{n} \log(\mathbb{O}(\mathbf{X}_i; \theta_j))$. Computing the log-space version of the BFF test statistics then leads to

$$\log(\tau(\mathcal{D}; \theta_0)) = \psi^0 - \log\left(\frac{1}{m}\sum_{j=1}^{m} \exp^{\psi_j}\right) = \psi^0 + \log(m) - \log\left(\sum_{j=1}^{m} \exp^{\psi_j}\right).$$

The above is computationally stable since the last term can be computed in a stable manner using any of the "log-sum-exp" implementations available (such as in SciPy, Virtanen et al. 2020).

Chapter 3

Validation of Approximate Likelihood and Emulator Models

Approximate likelihood models are usually employed as an intermediate step for downstream inference for posterior distribution. Prominent examples of this are the synthetic likelihood approaches in Bayesian inference (Wood, 2010; Meeds and Welling, 2014; Wilkinson, 2014; Price et al., 2018; Fasiolo et al., 2018; Picchini et al., 2020). Likelihood estimation has also been used in frequentist statistics, most notably in the discovery of the Higgs Boson (Aad et al., 2012) by estimating the likelihood of summary statistics as proposed by Diggle and Gratton (1984). Indeed, an accurate estimation of the likelihood is key for the validity of the downstream inference results. However, this is not the only setting in which likelihood approximations play a pivotal role. There is a growing number of disciplines where accurate analyses require highly realistic and computationally intensive simulations. In such cases, it may not be feasible to repeatedly generate new simulations at different parameter settings as generally required by ABC methods. Instead, a common practice is to run the simulator only for a few points in parameter space, in a format of batches or ensembles, where an *ensemble* is a collection of multiple realizations (e.g., corresponding to different initial conditions) of the same physical model (i.e. they all share the same θ). For example, modern climate and weather forecasting models (e.g., CESM (Hurrell et al., 2013)) often incorporate complex representations of the atmosphere, ocean, land, ice, etc, on fine spatial and temporal resolutions across the entire world. These models are commonly run as an ensemble of dynamical simulations with different initial conditions, where each simulation can take weeks to compile on supercomputer clusters (see (Baker et al., 2015a; Kay et al., 2015) and references within). Similarly, cosmological N-body simulations, which compute gravity between particle pairs, are equally costly and often either created at a fixed cosmology (parameter value θ) (Abbott et al., 2016; Hildebrandt et al., 2017), or on a sparse grid of a few carefully chosen parameter values (Kacprzak et al., 2016; Gupta et al., 2018).

Given the above scenario, a solution to make inference feasible is to replace the computationally expensive simulator with a faster *emulator* model that can speed up probabilistic modeling by several orders of magnitude. If the goal is to infer parameters θ of interest, then these emulators often forward-generate $\mathbf{x}$ given θ, thereby providing an explicit approximation to the likelihood.[*] Typically, machine learning-based LFI models are assessed by computing built-in loss functions (e.g., Kullback-Leibler divergences in autoregressive flows.) Such loss functions however only return a relative measure of performance rather than a goodness-of-fit to simulated data; they do not answer the question “Should we keep searching for better estimates for this problem or is our fit good enough?”. Thus, an important challenge is that of *validation*: determining whether an approximate likelihood or emulator model reproduces to the extent possible the targeted simulations in distribution. If the model is deemed inadequate, then the question of *diagnostics* becomes relevant. That is, pinpointing “how” and “where” the emulator differs from the simulator in a potentially high-dimensional feature space across different parameters; thereby providing valuable information for further improvements of the emulator, and insights on which simulations to run given a fixed budget. We propose a framework that can answer the following questions in a statistically rigorous way:

1. **if** one needs to improve emulators for reliable inference from observed data, i.e., whether the difference between the “truth” and the approximation learned with the existing train data is statistically significant; this question is answered by our global procedure (see Figure 3.4, left);

[*]Throughout this chapter we will use the terms emulator and approximate likelihood interchangeably to denote generative models that directly imply a relationship between observable data $\mathbf{x}$ and parameters θ.

2. **where** in parameter space one, if needed, should propose the next batch of simulations; this question is answered by our local procedure (Section 3.1.2) and provides insights as to which simulations to run given a fixed budget; and

3. **how** the distributions of emulated and high-resolution simulated data may differ in a potentially high-dimensional feature space; this question is answered by our regression test (see Section 3.1.3 and Figure 3.4, right) and offers valuable information as to what types of observations are under- or over-represented by the emulator and whether such differences are statistically significant. Such insights can guide decisions as to whether it is necessary to improve the emulator model or generate more simulations.

We use a new regression-based two-sample test Kim et al. (2016, 2019) to first compare the simulator and emulator models locally, i.e., at fixed parameters; these local tests are then aggregated into a "global" goodness-of-fit test that is statistically consistent (see Theorem 5). Our framework can adopt any machine-learning regression method to handle different structures in high-dimensional data. As Theorem 6 and Figure 3.2 show, this property translates to *high power* (for a fixed computational budget) under a variety of practical scenarios.

3.1 Validation via Local and Global Regression Test

Our validation approach compares samples from the simulator with samples from the emulator, and can detect local discrepancies for a given parameter setting $\theta_0 \in \Theta$ as well as global discrepancies across parameter settings in Θ. The validation procedure is as follows: For each $\theta_0 \in \Theta$, we first test the null hypothesis $H_0 : \widehat{\mathcal{L}}(\mathbf{x}; \theta_0) = \mathcal{L}(\mathbf{x}; \theta_0)$ for all $\mathbf{x} \in \mathcal{X}$. This *local test* (Algorithm 7) compares output from the approximate likelihood/emulator model with a "test sample" from the simulator/true likelihood (the latter sample can be a held-out subset of a pre-generated ensemble at θ_0 which has not been used to fit $\widehat{\mathcal{L}}(\mathbf{x}; \theta)$). A challenging problem is how to perform a two-sample test that is able to handle different types of data $\mathbf{x}$, and which in addition informs us on how two samples differ in feature space $\mathcal{X}$; in Section 3.1.2 and Algorithm 9 we propose a new *regression test* that addresses both these questions. After the two-sample comparisons, we combine local assessments into a *global test* (Algorithm 8) for checking if $\widehat{\mathcal{L}}(\mathbf{x};\theta) = \mathcal{L}(\mathbf{x};\theta)$ for all $\theta \in \Theta$. The essence of the global test is to pool p-values which, under the null hypothesis, are uniform. Unlike many previous works on pooling p-values for multiple testing (e.g., Lorenz et al. (2016)), the p-values in Algorithm 8 are independent by construction.

The next section provides theoretical guarantees that the global test for our LFI setting is indeed consistent. These results apply for any sampling/weighting scheme $r(\theta)$ over Θ in Algorithm 8, and for any consistent local test in Algorithm 7.

Algorithm 7 Local Test for Fixed θ

Input: parameter value θ_0, two-sample testing procedure, number of draws from the true model, $n_{\text{sim},0}$ and from the estimated model, $n_{\text{sim},1}$

Output: p-value p_{θ_0} for testing if $L(\mathbf{x};\theta_0) = \widehat{L}(\mathbf{x};\theta_0)$ for every $\mathbf{x} \in \mathcal{X}$

```
1: Sample $\mathcal{S}_0 = \{\mathbf{X}_1^{\theta_0}, \dots, \mathbf{X}_{n_{\text{sim},0}}^{\theta_0}\}$ from $\mathcal{L}(\mathbf{x};\theta_0)$.
2: Sample $\mathcal{S}_1 = \{\mathbf{X}_1^{*}, \dots, \mathbf{X}_{n_{\text{sim},1}}^{*}\}$ from $\widehat{\mathcal{L}}(\mathbf{x};\theta_0)$.
3: Compute $p$-value $p_{\theta_0}$ for the comparison between $\mathcal{S}_0$ and $\mathcal{S}_1$.
4: return $p_{\theta_0}$
```

Algorithm 8 Global Test Across $\theta \in \Theta$

Input: reference distribution $r(\theta)$, B, uniform testing procedure (e.g. Kolmogorov-Smirnoff, Cramér-von Mises)

Output: p-value p for testing if $L(\mathbf{x};\theta) = \widehat{L}(\mathbf{x};\theta)$ for every $\mathbf{x} \in \mathcal{X}$ and $\theta \in \Theta$

```
1: for $i \in \{1, \dots, B\}$ do
2:     sample $\theta_i \sim r(\theta)$
3:     compute $p_{\theta_i}$ using Algorithm 7
4: end for
5: Compute $p$-value $p$ for testing if $\{p_{\theta_i}\}_{i=1}^{B}$ has a uniform distribution.
6: return $p$
```

3.1.1 Theoretical Guarantees for Global Test

Here we provide sufficient assumptions for the global test to be statistically consistent; i.e., to be able to detect a misspecified distribution (as in Example 1) for large sample sizes.

Definition 1. *Define $\mathbb{D}_{B,n_{sim}} = \{p_{\theta_1}^{n_{sim}}, \dots, p_{\theta_B}^{n_{sim}}\}$, where $p_{\theta_1}^{n_{sim}}, \dots, p_{\theta_B}^{n_{sim}}$ are the p-values obtained by Algorithm 7 using $n_{sim,1} = n_{sim,2} = n_{sim}$, and $\theta_1, \dots, \theta_B \overset{i.i.d.}{\sim} r(\theta)$. Let $S(\mathbb{D}_{B,n_{sim}})$ be the test statistic for the global test. Also, denote by $S(\mathbb{U}_B)$ the test statistic when $\mathbb{U}_B = (U_1, \dots, U_B)$, with $U_1, \dots, U_B \overset{i.i.d.}{\sim} U(0,1)$.*

Assumption 8. *Let* $D = \left\{\theta : \mu_{\widehat{\mathcal{L}}(\cdot;\theta)} \neq \mu_{\mathcal{L}\cdot(\theta)}\right\}$, *where* $\mu_{\widehat{\mathcal{L}}(\cdot;\theta)}$ $(\mu_{\mathcal{L}(\cdot;\theta)})$ *is the measure over* $\mathcal{X}$ *induced by* $\mathcal{L}(\cdot;\theta)$ $(\widehat{\mathcal{L}}(\cdot;\theta))$. *Assume that* $\mu_r(D) > 0$, *where* μ_r *is the measure over* Θ *induced by* $r(\theta)$.

Assumption 9. *Assume that if* $\theta_1 \in D$, *then the local test is such that* $p_{\theta_1}^{n_{sim}} \xrightarrow[n_{sim}\longrightarrow\infty]{\mathbb{P}} 0$. *Moreover, if* $\theta_1 \notin D$, *then the local test is such that* $p_{\theta_1}^{n_{sim}} \sim U(0,1)$.

Assumption 10. *For every* $0 < \alpha < 1$, *the test statistic* S *is such that* $F^{-1}_{S(\mathbb{U}_B)}(1-\alpha) \xrightarrow{B\longrightarrow\infty}$ 0.

Assumption 11. *Under Assumptions 8 and 9, there exists* $a > 0$ *such that the test statistic* S *satisfies* $S(\mathbb{D}_{B,n_{sim}}) \xrightarrow[B,n_{sim}\longrightarrow\infty]{\mathbb{P}} a$.

Assumption 8 states that the set of parameter values where the likelihood function is incorrectly estimated has positive mass under the reference distribution. Assumption 9 states that the test chosen to perform the local comparisons is statistically consistent and that its p-value has uniform distribution under the null hypothesis. Assumptions 10 and 11 state that the test statistic for the global comparison in step 5 of Algorithm 8 is statistically consistent, i.e., (i) it approaches zero under the null hypothesis when B increases, and (ii) it converges to a positive number if the null hypothesis is false. Under these four assumptions, we can guarantee statistical consistency.

Theorem 5. *Let* ϕ *be an* α*-level testing procedure based on the global test statistic* S. *If the likelihood estimate and the local and global test statistics are such that Assumptions 8–11 hold, then*

$$\mathbb{P}\left(\phi_S(\mathbb{D}_{B,n_{sim}}) = 1\right) \xrightarrow{B,n_{sim}\longrightarrow\infty} 1$$

Corollary 3. *Under Assumptions 8 and 9, the global tests for comparing likelihood models based on Kolmogorov-Smirnoff and Cramér-von Mises statistics are statistically consistent.*

Lemma 1. *Let* $\widehat{F}_{\mathbb{D}_{B,n_{sim}}}$ *be the empirical cumulative distribution of the p-values in* $\mathbb{D}_{B,n_{sim}}$,

$$KS(\mathbb{D}_{B,n_{sim}}) = \sup_{0\leq z\leq 1} |\widehat{F}_{\mathbb{D}_{B,n_{sim}}}(z) - z|,$$

be the Kolmogorov-Smirnoff test statistic and

$$CVM(\mathbb{D}_{B,n_{sim}}) = \int_0^1 \left(\widehat{F}_{\mathbb{D}_{B,n_{sim}}}(z) - z\right)^2 dz$$

be the Cramér-von Mises test statistic. Both KS and CVM satisfy Assumptions 10 and 11.

Proof of Lemma 1. Let $U \sim U(0,1)$. From the law of large numbers,

$$\text{KS}(\mathbb{U}_B) = \sup_{0 \leq z \leq 1} |\widehat{F}_{\mathbb{U}_B}(z) - z| \xrightarrow[B \longrightarrow \infty]{\text{a.s.}} \sup_{0 \leq z \leq 1} |\mathbb{P}(U \leq z) - z| = 0,$$

which proves the first statement of the theorem. Similarly, for every $n_{\text{sim}} \in \mathbb{N}$,

$$\text{KS}(\mathbb{D}_{B,n_{\text{sim}}}) = \sup_{0 \leq z \leq 1} |\widehat{F}_{\mathbb{D}_{B,n_{\text{sim}}}}(z) - z| \xrightarrow[B \longrightarrow \infty]{\text{a.s.}} \sup_{0 \leq z \leq 1} |\mathbb{P}(p_{\theta_1}^{n_{\text{sim}}} \leq z) - z|. \tag{3.1}$$

Now, Under Assumption 9, for every $\theta_1 \in D$,

$$\mathbb{P}(p_{\theta_1}^{n_{\text{sim}}} \leq z|\theta_1) \xrightarrow{n_{\text{sim}} \longrightarrow \infty} 1$$

uniformly over $z \in (0,1)$. Thus, under Assumption 8, for every $0 < \epsilon_z < 1 - z$, there exists $n_{\text{sim}} \in \mathbb{N}$ such that, for every $n'_{\text{sim}} > n_{\text{sim}}$,

$$\begin{aligned}
\mathbb{P}(p_{\theta_1}^{n'_{\text{sim}}} \leq z) &= \mathbb{P}(p_{\theta_1}^{n'_{\text{sim}}} \leq z|\theta_1 \in D)\mathbb{P}(\theta_1 \in D) + \quad \mathbb{P}(p_{\theta_1}^{n'_{\text{sim}}} \leq z|\theta_1 \notin D)\mathbb{P}(\theta_1 \notin D) \\
&\geq (1 - \epsilon_z)\mathbb{P}(\theta_1 \in D) + z\mathbb{P}(\theta_1 \notin D) \\
&= (1 - \epsilon_z + z - z)\mathbb{P}(\theta_1 \in D) + z\mathbb{P}(\theta_1 \notin D) \qquad (3.2) \\
&= (1 - \epsilon_z - z)\mathbb{P}(\theta_1 \in D) + z \qquad (3.3)
\end{aligned}$$

It follows from Equations 3.1 and 3.2 and by taking $\epsilon_z = (1-z)/2$ that

$$\begin{aligned}
\sup_{0 \leq z \leq 1} |\mathbb{P}(p_{\theta_1}^{n'_{\text{sim}}} \leq z) - z| &\geq \sup_{0 \leq z \leq 1} (1 - \epsilon_z - z)\mathbb{P}(\theta_1 \in D) \\
&\geq \mathbb{P}(\theta_1 \in D) \sup_{0 \leq z \leq 1} \frac{(1-z)}{2} = \frac{\mathbb{P}(\theta_1 \in D)}{2},
\end{aligned}$$

and hence

$$\lim_{n'_{\text{sim}} \longrightarrow \infty} \sup_{0 \leq z \leq 1} |\mathbb{P}(p_{\theta_1}^{n'_{\text{sim}}} \leq z) - z| \geq \frac{\mathbb{P}(\theta_1 \in D)}{2} > 0,$$

which concludes the proof for the KS statistic. The proof for the CVM statistic is analogous.

□

Proof of Theorem 5. Assumption 9 implies that ϕ_S is such that

$$\phi_S(\mathbb{D}_{B,n_{\text{sim}}}) = 1 \iff S(\mathbb{D}_{B,n_{\text{sim}}}) \geq F^{-1}_{S(\mathbb{U}_B)}(1-\alpha).$$

It follows that

$$\begin{aligned}\mathbb{P}\left(\phi_S(\mathbb{D}_{B,n_{\text{sim}}}) = 1\right) &= \mathbb{P}\left(S(\mathbb{D}_{B,n_{\text{sim}}}) - F^{-1}_{S(\mathbb{U}_B)}(1-\alpha) \geq 0\right) \\ &\geq \mathbb{P}\left(|S(\mathbb{D}_{B,n_{\text{sim}}}) - a - F^{-1}_{S(\mathbb{U}_B)}(1-\alpha)| \leq a\right) \xrightarrow{B, n_{\text{sim}} \longrightarrow \infty} 1,\end{aligned}$$

where the last line follows from Assumptions 10 and 11. □

Proof of Corollary 3. It follows directly from Theorem 5 and Lemma 1. □

3.1.2 Two-Sample Test via Regression

Traditional approaches to comparing two distributions (Thas, 2010) often do not easily generalize to high-dimensional and non-Euclidean data. More recent non-parametric extensions (see Hu and Bai (2016) for a review), e.g., maximum mean discrepancy (MMD, Gretton et al. 2012), energy distance (ED, Székely and Rizzo 2004), divergence (Sugiyama et al., 2011; Kanamori et al., 2012), mean embedding (Chwialkowski et al., 2015; Jitkrittum et al., 2016) and classification accuracy tests (Kim et al., 2021; Lopez-Paz and Oquab, 2017) have shown to have power in high dimensions against some alternatives, specifically location and scale alternatives. These methods, however, only provide a binary answer of the form "reject" or "fail to reject" the null hypothesis. Here we propose a new regression-based approach to two-sample testing that can adapt to any structure in $\mathcal{X}$ where there is a suitable regression method; Theorem 6 relates the power of the test to the Mean Integrated Squared Error (MISE) of the regression. Moreover, the regression test can detect and describe local differences (beyond the usual location and scale alternatives) in $\widehat{\mathcal{L}}(\mathbf{x}; \theta_0)$ and

$\mathcal{L}(\mathbf{x};\theta_0)$ in feature space $\mathcal{X}$. We briefly describe the method below; see Kim et al. (2019) for theoretical details, and see Section 3.2 for examples based on random forest regression.

Let P_0 be the distribution over $\mathcal{X}$ induced by $\mathcal{L}(\mathbf{x};\theta_0)$ and let P_1 be the distribution over $\mathcal{X}$ induced by $\widehat{\mathcal{L}}(\mathbf{x};\theta_0)$. Assume that P_0 and P_1 have density functions f_0 and f_1 relative a common dominating measure. By introducing a random variable $Y \in \{0,1\}$ that indicates which distribution an observation belongs to, we can view f_0 and f_1 as conditional densities $f(\mathbf{x}|Y=0)$ and $f(\mathbf{x}|Y=1)$. The local null hypothesis is then equivalent to the hypothesis $H_0 : f_0(\mathbf{x}) = f_1(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}_0 := \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) > 0\}$, which in turn is equivalent to

$$H_0:\ \mathbb{P}(Y=1|\mathbf{X}=\mathbf{x}) = \mathbb{P}(Y=1), \quad \text{for all } \mathbf{x} \in \mathcal{X}_0.$$

We test H_0 against the alternative $H_1:\ \mathbb{P}(Y=1|\mathbf{X}=\mathbf{x}) \neq \mathbb{P}(Y=1), \quad \text{for some } \mathbf{x} \in \mathcal{X}_0.$

By the above reformulation, we have converted the problem of two-sample testing to a *regression* problem. Depending on the choice of method for estimating the regression function $m(\mathbf{x}) = \mathbb{P}(Y=1|\mathbf{X}=\mathbf{x})$, we can adapt to nontraditional data settings involving mixed data types and various structures. More specifically, let $\widehat{m}(\mathbf{x})$ be an estimate of $m(\mathbf{x})$ based on the sample $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, and let $\widehat{\pi}_1 = \frac{1}{n}\sum_{i=1}^n I(Y_i = 1)$. We define our test statistic as

$$\widehat{\mathcal{T}} = \frac{1}{n}\sum_{i=1}^{n} \left(\widehat{m}(\mathbf{X}_i) - \widehat{\pi}_1\right)^2 . \tag{3.4}$$

Note that the difference $|\widehat{m}(\mathbf{x}) - \widehat{\pi}_1|$ *for each particular value of* $\mathbf{x} \in \mathcal{X}$ also provides information on how well the emulator fits the simulator locally *in feature space*; high values indicate a poor fit. To keep our framework as general as possible, we use a permutation procedure (Algorithm 9) to compute p-values; one could alternatively use a goodness-of-fit test via Monte Carlo sampling (see Section 3.4.1).

Theorem 6 shows that if $\widehat{m}$, the chosen regression estimator, has a small MISE, the power of the test is large over a wide region of the alternative hypothesis. What this means in practice is that we should choose a regression method that predicts the "class membership" Y well.

Algorithm 9 Two-Sample Regression Test via Permutations

```
Input: two i.i.d. samples $\mathcal{S}_0$ and $\mathcal{S}_1$ from distributions with resp. densities $f_0$ and $f_1$; number of permutations $M$; a regression method $\widehat{m}$
Output: $p$-value for testing if $f_0(\mathbf{x}) = f_1(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$
1: Define an augmented sample $\{\mathbf{X}_i, Y_i\}_{i=1}^n$, where $\{\mathbf{X}_i\}_{i=1}^n = \mathcal{S}_0 \cup \mathcal{S}_1$, and $Y_i = I(\mathbf{X}_i \in \mathcal{S}_1)$.
2: Calculate the test statistic $\widehat{\mathcal{T}}$ in Equation 3.4.
3: Randomly permute $\{Y_1, \ldots, Y_n\}$. Refit $\widehat{m}$ and calculate the test statistic on the permuted data.
4: Repeat the previous step $M$ times to obtain $\left\{\widehat{\mathcal{T}}^{(1)}, \ldots, \widehat{\mathcal{T}}^{(M)}\right\}$.
5: Approximate the permutation $p$-value by $p = \frac{1}{M+1}\left(1 + \sum_{m=1}^M I(\widehat{\mathcal{T}}^{(m)} > \widehat{\mathcal{T}})\right)$.
6: return $p$
```

Theorem 6. *Suppose that the regression estimator $\widehat{m}(\mathbf{x})$ is a linear smoother satisfying*

$$\sup_{m \in \mathcal{M}} \mathbb{E} \int_{\mathcal{X}} (\widehat{m}(\mathbf{x}) - m(\mathbf{x}))^2 \, dP_X(\mathbf{x}) \leq C_0 \delta_n,$$

where C_0 is a positive constant, $\delta_n = o(1)$, $\delta_n \geq n^{-1}$, and $\mathcal{M}$ is a class of regressions $m(\mathbf{x})$ containing constant functions. Let t_α^* be the upper α quantile of the permutation distribution of the test statistic $\widehat{\mathcal{T}}'$ on validation data from sample splitting[†]*. Then for any $\alpha, \beta \in (0, 1/2)$ and n sufficiently large, there exists a universal constant C_1 such that*

$$\text{Type I error: } \mathbb{P}_0\left(\widehat{\mathcal{T}}' \geq t_\alpha^*\right) \leq \alpha,$$

$$\text{Type II error: } \sup_{m \in \mathcal{M}(C_1\delta_n)} \mathbb{P}_1\left(\widehat{\mathcal{T}}' < t_\alpha^*\right) \leq \beta$$

against the class of alternatives $\mathcal{M}(C_1\delta_n) := \left\{m \in \mathcal{M} : \int_{\mathcal{X}} (m(\mathbf{x}) - \pi_1)^2 \, dP_X(\mathbf{x}) \geq C_1 \delta_n\right\}$.

For proof of Theorem 6 we refer the reader to Kim et al. (2019).

3.1.3 Local Test in Feature Space

Our regression approach can be used to identify and visualize locally significant differences between two multivariate distributions P_0 and P_1 defined over a "feature space" $\mathcal{X}$; we denote samples from the respective distributions by $\mathcal{S}_0$ and $\mathcal{S}_1$. The goal would be to identify

[†]The proof assumes sample splitting where (for simplicity) half of the data is used to estimate the regression function and the other half is used to estimate the test statistic; i.e., $\widehat{\mathcal{T}}' = 2n^{-1} \sum_{i=n/2}^{n} (\widehat{m}(\mathbf{X}_i) - \widehat{\pi}_1)^2$, where $\widehat{m}$ is estimated using $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{n/2}, Y_{n/2})$.

with statistical confidence the regions in $\mathcal{X}$ which may be under- or over-represented in $\mathcal{S}_1$ (as compared to S_0). Our approach is provided in Algorithm 10 and provides p-values for testing the significance of difference in $|\widehat{m}(\mathbf{X}_j) - \widehat{\pi}_1|$, for a given point $\mathbf{X}_j$ in the test set. The key idea is to compute the following test statistic on each of the test points $\mathbf{X}_j$:

$$\widehat{\nu}(\mathbf{X}_j) = (\widehat{m}(\mathbf{X}_j) - \widehat{\pi}_1)^2,$$

and then obtain p-values using a permutation test for the distribution of $\widehat{v}$. An example of our local testing capabilities in provided in Figure 3.4, right, where S_0 are the true sample from the `CAMELUS` simulator and S_1 the sample from a fitted Gaussian model. Our techniques can also be used to validate and diagnose output from generative adversarial networks (GANs) and other so-called implicit generative models Mohamed and Lakshminarayanan (2016); e.g., this type of analysis could be relevant for recent GAN models of galaxy images (Ravanbakhsh et al., 2017) and weak lensing convergence maps (Mustafa et al., 2019).

Algorithm 10 Local Test in Feature Space

Input: i.i.d. training data from two populations $\{\mathbf{X}_i, Y_i\}_{i=1}^n$; testing data $\{\mathbf{X}_j\}_{j=1}^J$; number of permutations M; significance level α; a regression method $\widehat{m}$

Output: p-values $\{p_j\}_{j=1}^M$ for testing significance of difference $|\widehat{m}(\mathbf{X}_j) - \widehat{\pi}_1|$ for every test point

```
1: $\widehat{\pi}_1 = 1/n \sum_{i=1}^n Y_i$;
2: Train regression method $\widehat{m}$ on training data $\{\mathbf{X}_i, Y_i\}_{i=1}^n$;
3: Calculate the test statistics on each of the test points
       $\widehat{\nu}(\mathbf{X}_j) = (\widehat{m}(\mathbf{X}_j) - \widehat{\pi}_1)^2$;
4: for $k$ in $1, ..., M$ do
5:     Randomly permute $Y_1, ..., Y_n$ and train regression method on permuted data $\widehat{m}^{(k)}$;
6:     Calculate the test statistics on the permuted data $\{\widehat{\nu}^{(k)}(\mathbf{X}_j) = (\widehat{m}^{(k)}(\mathbf{X}_j) - \widehat{\pi}_1)^2\}_{j=1}^J$;
7: end for
8:
9: Approximate permutation p-values $p_j$ for every test point $\mathbf{X}_j$:
       $p_j = \frac{1}{M+1} \sum_{k=1}^M \left(1 + \mathbb{I}(\widehat{\nu}^{(k)}(\mathbf{X}_j) > \widehat{\nu}(\mathbf{X}_j))\right)$
10: Apply a multiple test procedure to control false discovery rate;
11: return $\{p_j\}_{j=1}^J$
```

3.2 Comparison with Traditional Consistency Checks

Up to now, popular approaches to simulation-based validation (Cook et al., 2006; Prangle et al., 2014; Talts et al., 2018) are valuable as consistency checks, but cannot always identify likelihood models that are clearly misspecified (see Section 3.2.1 for an example). Furthermore, as these tools were originally designed for checking posterior approximations in Bayesian models, they do not capture all aspects of the estimated likelihood, and therefore provide limited information on how to improve the estimates. In addition, there are also close connections between classification accuracy tests (Kim et al., 2021; Lopez-Paz and Oquab, 2017) and our regression test. The main difference lies in the test statistic: classification accuracy tests are based on "global" error rates. Hence classifier tests can tell whether two distributions are different (i.e. they are two-sample tests) but these tests do not *per se* identify locally significant differences between two distributions with statistical confidence; for that one needs to consider the regression or class-conditional probabilities $\mathbb{E}(Y|\mathbf{x}) = \mathbb{P}(Y = 1|\mathbf{x})$ (where Y here is the indicator function that $\mathbf{x}$ was generated by the emulator as opposed to the forward-simulator), which is the basis of our regression test statistic (Equation 3.4).

We next use two synthetic examples to illustrate the advantages of our global and local tests to state-of-the-art validation techniques, in terms of consistency and higher power respectively.

3.2.1 Global Test and Existing Diagnostic Tools

One key property of our global goodness-of-fit test is that it can detect any misspecified approximation of the likelihood function (Theorem 5). Diagnostic tools like the Posterior Quantiles (PQ) technique (Cook et al., 2006) and Simulation-Based Calibration (SBC Talts et al. 2018) are often used to validate approximate likelihood models (see, e.g., Papamakarios and Murray 2016) by checking whether a histogram of respective statistics (posterior quantiles and ranks) is close to uniform. However, these tests are sometimes not able to tell the difference between the true model and a clearly misspecified model as illustrated by the following toy example, where $\theta_i \sim \text{Gamma}(1, 1)$, $i = 1, \ldots, 500$, and $\mathbf{x} = x_1, \ldots, x_{1000}|\theta_i \sim \text{Beta}(\theta_i, \theta_i)$. The PQ test is based on the fact that, given a sample $\tilde{\theta}$

from the prior distribution, the posterior quantile

$$q(\tilde{\theta}) = \int f(\theta|\mathbf{x})\mathbb{I}(\theta < \tilde{\theta})d\theta$$

is uniformly distributed. Similarly, the SBC test relies on the fact that, given any ranking function $g(\theta)$ and a posterior sample $\{\theta_1, \ldots, \theta_L\}$, the rank

$$r\left(g(\theta_1), ..., g(\theta_L), g(\tilde{\theta})\right) = \sum_{l=1}^{L} \mathbb{I}(g(\theta_l) < g(\tilde{\theta}))$$

is uniformly distributed. Both PQ and SBC assess goodness-of-fit by checking if a histogram of respective statistics (posterior quantiles and ranks) is close to uniform.

Figure 3.1, left, shows the distribution of the statistics computed for PQ and SBC (along with confidence regions that describe what one would expect under uniformity) and the distribution of our local p-values (recall that our global test is based on testing whether the local p-values are uniformly distributed) for two different cases: In the top row, we consider a case where $\widehat{\mathcal{L}}(\mathbf{x};\theta) = \mathcal{L}(\mathbf{x};\theta)$. All tests pass the model, as they should. In the bottom row, we consider a case where $\widehat{\mathcal{L}}(\mathbf{x};\theta) \propto 1$, a poor approximation of the likelihood function (see Figure 3.1, right, for examples.). Our global regression test, which is based on uniformity of the local p-values, clearly rejects this model. PQ and SBC, on the other hand, cannot distinguish between the true likelihood and the misspecified model as these by construction have the same marginal distribution over θ in this toy example. Similar results (Schmidt et al., 2020) have been found for diagnostic tests of conditional density estimates when using quantities related to PQ and SBC (such as, PIT scores and QQ plots).

3.2.2 Local Test and Existing Goodness-of-Fit Tests

The power of our goodness-of-fit test will much depend on how we compare samples at fixed $\theta_0 \in \Theta$; that is, on how we test the local null hypothesis, $H_0 : \widehat{\mathcal{L}}(\mathbf{x};\theta_0) = \mathcal{L}(\mathbf{x};\theta_0)$ for every $\mathbf{x} \in \mathcal{X}$. An advantage of the regression approach (Algorithm 9) is that we can use any regression technique that efficiently explores the structure of the data at hand; the practical implications of Theorem 6 is that one should choose the regression method with the smallest MISE (a quantity that can be estimated from data) to attain a higher test

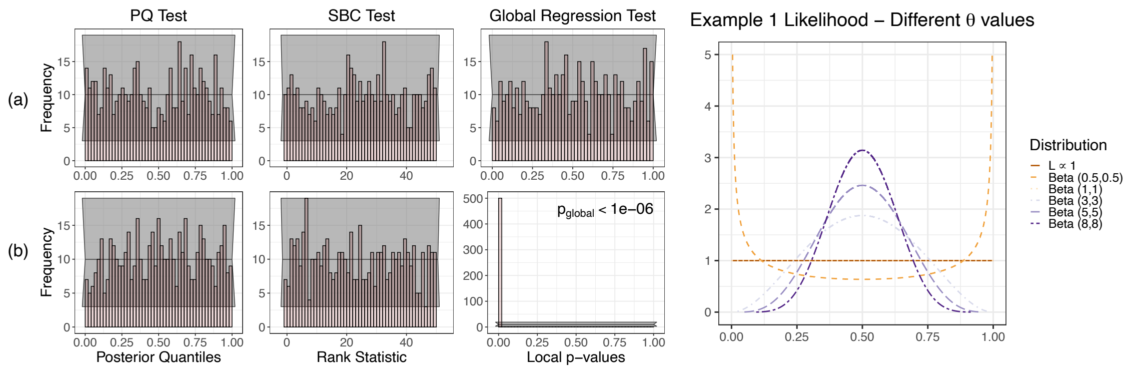


Figure 3.1: *Left*: Distribution of posterior quantiles, rank statistics and p-values for PQ, SBC and our global regression test, respectively, for (a) the true model, and (b) a clearly misspecified model. Only the global regression test correctly rejects the latter (bottom right plot). (The grey ribbon represents the 99% confidence interval for the test of uniformity used for PQ and SBC.) *Right*: The true likelihood for different values of the parameter θ, compared to the approximation $\widehat{\mathcal{L}}(\mathbf{x};\theta) \propto 1$. The approximation is clearly wrong when $\theta \neq 1$.

power (an unknown quantity). We illustrate these ideas with a synthetic example where $\mathbf{x} \in \mathbb{R}^D$, where D could be large. We consider three toy settings where the approximate likelihood and the true likelihood only differ in the first dimension — that is, we test against a sparse alternative; see Table 3.1 for details.

Settings	**True Likelihood** $\mathcal{L}(\mathbf{x};\theta)$	**Approx. Likelihood** $\widehat{\mathcal{L}}(\mathbf{x};\theta)$	Θ **Space**
(a) Bernoulli	$\text{Bern}(x_1;\theta)\prod_{d=2}^{D}\mathcal{N}(x_d;\theta,1)$	$\prod_{d=1}^{D}\mathcal{N}(x_d;\theta,1)$	$(0,1)$
(b) Scaling	$\mathcal{N}(x_1;0,\theta)\prod_{d=2}^{D}\mathcal{N}(x_d;0,1)$	$\prod_{d=1}^{D}\mathcal{N}(x_d;0,1)$	$(0,1)$
(c) Mixture of Gaussians	$f_m(x_1;\theta,1)\prod_{d=2}^{D}\mathcal{N}(x_d;0,1)$, where $f_m(\theta,1)=1/2\mathcal{N}(-\theta,1)+1/2\mathcal{N}(\theta,1)$	$\prod_{d=1}^{D}\mathcal{N}(x_d;0,1)$	$(-5,5)$

Table 3.1: The three toy settings. In each setting, the true and approximate likelihood differ only in the first dimension, x_1. ($\mathcal{N}(x;\mu,\sigma^2)$ is a 1D Gaussian with mean μ and variance σ^2; $\text{Bern}(x;\theta)$ is a Bernoulli with parameter θ.)

For each $\theta \in \Theta$, we compute a local p-value by comparing samples of size $n = 100$ from $\mathcal{L}(\mathbf{x};\theta)$ and $\widehat{\mathcal{L}}(\mathbf{x};\theta)$, respectively (Algorithm 7). This procedure is repeated 100 times to estimate the power function. We apply the local test for three different test statistics; namely: (i) the test statistic in Equation 3.4 using random forest (RF) or nearest neighbor regression (NN), (ii) the MMD test statistic (Gretton et al., 2012, Eq. 5) with a Gaussian kernel, and (iii) the energy test statistic (Baringhaus and Franz, 2004; Szákely and Rizzo,

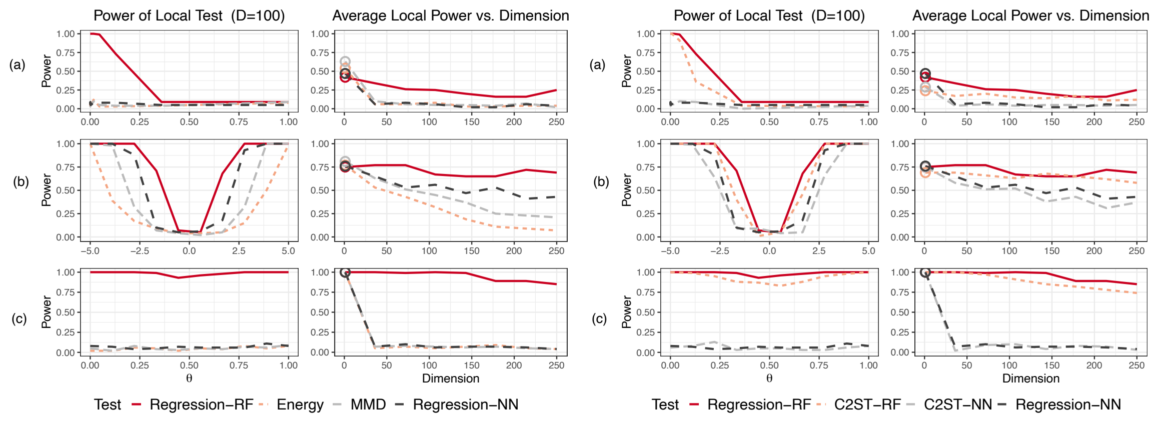


Figure 3.2: *Left:* Local test power shown in the left column as a function of θ at D=100, and shown in the right column as a function of the dimension D (averaged over θ) for the (a) Bernoulli, (b) Scaling, and (c) Mixture of Gaussians case. Note that distance-based tests are more powerful at $D = 1$ (highlighted with circles in the right column), but their power is severely affected with increasing dimension. Our RF regression test achieves higher power for large D by leveraging the advantages of random forest regression in high-dimensional settings with sparse structure. *Right:* Test power at $D = 100$ (left column) and as a function of dimension D (right column) in the same Example 2 settings, i.e., for (a) Bernoulli, (b) Scaling and (c) Mixture of Gaussians. We include the results for our regression test with random forests (RF) and nearest neighbors (NN), as well as the corresponding results using the classification accuracy test of Lopez-Paz and Oquab (2017) with RF and NN (labeled as C2ST-RF and C2ST-NN, respectively).

2004, Eq. 5) using the Euclidean norm. Figure 3.2, left, shows how the power function varies with θ at dimension $D = 100$ (left column) and how the power, averaged over θ, varies with D (right column) for each setting. When $D = 1$ (highlighted with circles in the right column) distance-based tests based on RF yield higher power, but their performance quickly degrades with increasing D. On the other hand, our RF regression test is able to achieve higher power in high-dimensional settings by leveraging some advantages of random forest regression (as shown by the red curves); such as, the ability to select features, and the ability to tell discrete versus continuous distributions apart. For instance, in the Bernoulli case (top row, a) our regression test has higher power for small values of θ, which is when the distribution of the first coordinate is almost degenerate at 0.

Moreover, the practical implications of Theorem 6 are that for a two-sample test via regression one should base the test on the regression method with the smallest mean integrated squared error (MISE) so as to achieve a more powerful test. Table 3.2 illustrates this for the three settings random forest achieves a smaller MISE than nearest neighbor (NN) regression across all settings and, as Figure 3.2 shows, it also consistently attains a higher power.

Setting / Regression Method	*Random Forest*	*NN*
(a) Bernoulli	0.19	0.73
(b) Scaling	0.35	2.31
(c) Mixture of Gaussians	0.27	1.64

Table 3.2: Integrated mean squared error (MISE) for regression methods used for two-sample testing in Figure 3.2. Random forest has the smallest MISE in regression; it also yields the test with highest power, as implied by Theorem 6.

As mentioned above, classifier two-sample testing methods have also been used for two-sample testing by dichotomizing the regression function and using the classification accuracy as a test statistic. Such dichotomization might result in a loss of power with respect to the respective regression test in certain settings (for more examples, see Kim et al. 2019). In Figure 3.2, right, we consider the same settings as in Table 3.1, but now also computing the power of the classification accuracy test from Lopez-Paz and Oquab (2017) for both random forest and nearest neighbor classification. The regression test achieves comparable results across the different settings, providing slight improvements in some cases, e.g., with respect to the local power at $D = 100$ (left column). Note that our global procedure can incorporate classification accuracy tests as well, but would then not be able to identify locally significant differences in feature space as in Section 3.1.3.

3.3 Application: Constraining Cosmological Parameters with Weak-Lensing Peak Counts

In this section we focus on *validating* approximate likelihood models for cosmological parameter inference with weak lensing peak counts. Weak lensing (WL) is a gravitational deflection effect of light by matter in the Universe that causes distortion in projected images of distant galaxies along lines of sight. We can use this effect to estimate parameters of the ΛCDM cosmological model, the most well-supported model within Big Bang cosmology. In particular we can estimate the dark matter density Ω_m and its clumpiness σ_8 through *peak counts*: the number of local maxima in the WL convergence map (a 2D image) binned by the value of the peak (Dietrich and Hartlap, 2010).

In our data example, we use the `CAMELUS` simulator (Lin and Kilbinger, 2015) to generate peaks. We provide results and insights with data obtained from the `CAMELUS` simulator, comparing two parametric models, a Gaussian and a Poisson model, with a conditional masked autoregressive flow (MAF; Papamakarios et al. 2017), again discretized to reflect the integer-valued data. Our choice of parametric model is motivated by the fact that the Gaussian model with a fixed covariance and varying mean is the current state-of-the-art in cosmological parameter inference (Kacprzak et al., 2016). We consider a 2D parameter space over $\theta = (\Omega_m, \sigma_8)$ and design a grid of 50 different cosmologies θ around a fiducial (probable) cosmology θ_0 (see Appendix D). For each θ-value, we simulate a batch of WL maps ($n_{\text{train}} = 200$, $n_{\text{sim}} = 200$). The peak count data (i.e. histogram of peak intensities in each map) is a vector $\mathbf{x} \in \mathbb{N}^D$ where $D = 7$ is the number of bins. Figure 3.3 shows the grid of 50 parameters settings $\theta = (\Omega_m, \sigma_8)$ which we use for the `CAMELUS` batch simulations. The blue shaded region represents the parameter regions from which the parameters are sampled around the fiducial cosmology θ_0 (indicated by a red diamond).

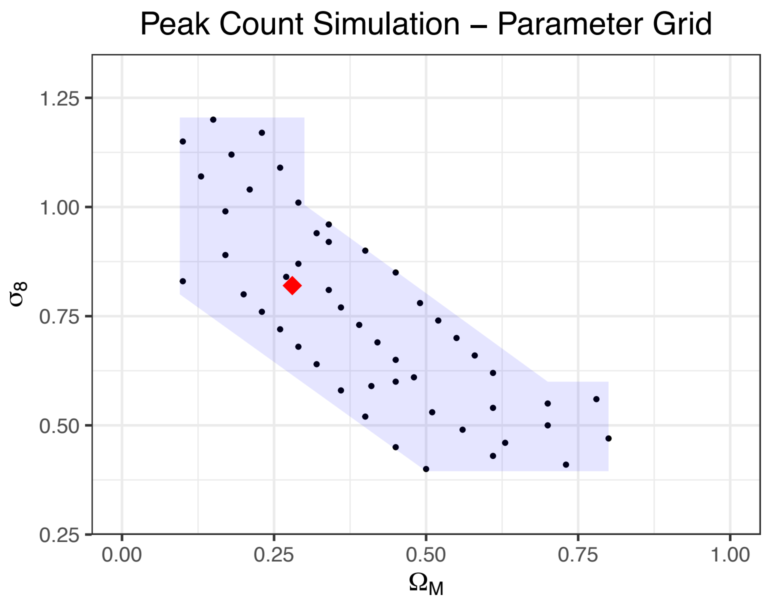


Figure 3.3: Location of the 50 parameter values for the peak count data simulations using `CAMELUS`, where the blue region indicates the parameter range values and the red diamond indicates the fiducial point θ_0.

To assess models, we first compute the Kullback-Leibler (KL) divergence loss for the $n_{sim} = 200$ test simulations at each θ. The KL divergence for model comparison is estimated by:

$$KL(\mathcal{L}, \widehat{\mathcal{L}}) = -\mathbb{E}\left[\log\left(\frac{\widehat{\mathcal{L}}(\mathbf{x};\theta)}{\mathcal{L}(\mathbf{x};\theta)}\right)\right] = -\mathbb{E}\left[\log\left(\widehat{\mathcal{L}}(\mathbf{x};\theta)\right)\right] + K \approx -\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n_j}\log\left(\widehat{\mathcal{L}}(\mathbf{x}_{ij};\theta_j)\right) + K$$

where K does not depend on $\widehat{\mathcal{L}}$; $\{\theta_j\}_{j=1}^{m}$ with $m = 50$ denotes the parameters used by the simulator; $\{\mathbf{x}_{ij}\}_{i=1}^{n_j}$ (with $n_j = 200$ for all θ_j) denotes the test simulations at θ_j; and $\sum_{j=1}^{m} n_j = n$ is the total number of test simulations. According to the KL loss, the Gaussian model performs best (Figure 3.4, left); however, these are only relative comparisons. We now use a RF regression test to find out whether the Gaussian model actually fits the simulated data well. As indicated in Figure 3.4 (left, top row), the local tests for the Gaussian model reject the null hypothesis $\widehat{\mathcal{L}}(\mathbf{x};\theta) = \mathcal{L}(\mathbf{x};\theta)$ at every θ; thus the global hypothesis is also rejected. The Poisson and MAF models are rejected by the global test as well but have a more uniform-looking distribution of local p-values. Now if we increase the the number of train simulations to $n_{\text{train}} = 500$ (while holding $n_{\text{sim}} = 200$ fixed), the fitted MAF model passes the global test whereas the Gaussian and Poisson models still do not as indicated by the bottom row (these qualitative results stay the same for $n_{\text{train}} = 5000$).

Finally, our local regression tests can provide insights into *how* the two distributions $\widehat{\mathcal{L}}(\mathbf{x};\theta)$ and $\mathcal{L}(\mathbf{x};\theta)$ differ in feature space $\mathcal{X}$; more specifically, by evaluating how the estimate of the regression function $\widehat{m}(\mathbf{x})$ in Equation 3.4 varies with $\mathbf{x}$ for a fixed θ (a significant difference $|\widehat{m}(\mathbf{x}) - \widehat{\pi}_1|$ is an indication that the model is not well estimated at that location in feature space) We illustrate such an analysis for our fitted Gaussian model for $n_{\text{train}} = 200$ and $\theta = \theta_0$. According to the RF regression used to construct our test statistic, the most influential variables correspond to bins with low counts. In Figure 3.4, right, we visualize the fit on such a bin (variable x_7) by a partial dependence plot (which shows the marginal effect of this variable on $\widehat{m}(\mathbf{x})$ Friedman (2001)). On the x-axis, we mark the locations where the difference $|\widehat{m}(\mathbf{x}) - \widehat{\pi}_1|$ is statistically significant according to a joint analysis in 7 dimensions (see Algorithm 10 for details). These locations coincide with integer values of x_7, showing that the regression test is distinguishing between the

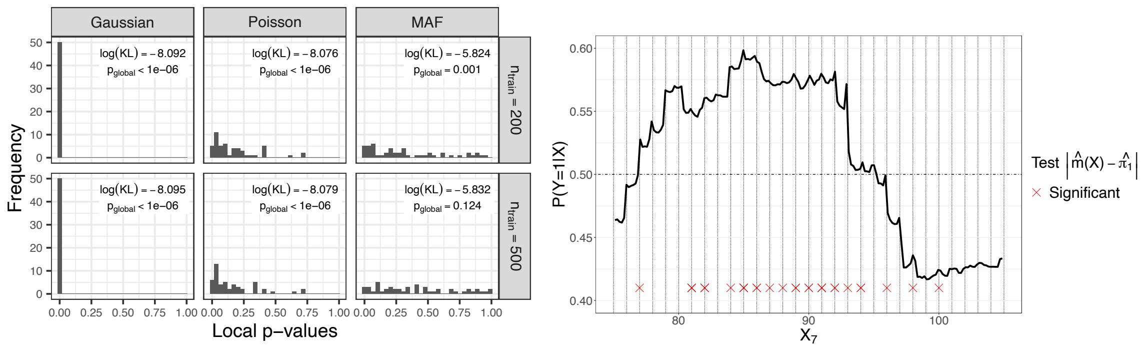


Figure 3.4: *Left panel*: Local goodness-of-fit for peak-count data with $n_{\text{train}} = 200$ (top row) and $n_{\text{train}} = 500$ (bottom row). Although the Gaussian model is achieving the lowest KL divergence, the estimates are rejected at almost all θ; Poisson and MAF perform better (more uniform-looking distributions of local p-values) but only MAF passes the global test at $n_{\text{train}} = 500$. *Right panel*: Partial dependence plot for variable x_7 (low count bin) for Gaussian model at $n_{\text{train}} = 200$. The red crosses on the x-axis represent the locations where the difference $|\widehat{m}(\mathbf{x}) - \widehat{\pi}_1|$ is statistically significant according to a joint analysis in 7 dimensions; these locations coincide with integer values of x_7 and indicate that the regression test is distinguishing between the discrete true distribution of counts and the fitted continuous Gaussian distribution.

discrete true distribution for bin counts and the fitted continuous Gaussian distribution (these results also explain why the Poisson model may fare better).

3.4 Extensions

In this section we present two direct extensions of our presented approach. We first present an alternative approach to goodness-of-fit testing when sampling from the simulator is expensive but sampling from the emulator is not. The key idea is that one can instead of the two-sample permutation test in Algorithm 9 perform a goodness-of-fit test via repeated Monte Carlo sampling from the emulator. Secondly, we show how a validated approximate likelihood model can be used for frequentist inference to construct p-values and confidence regions.

3.4.1 Testing with an Inexpensive Approximate Likelihood Model

If the total number of test simulations from $\mathcal{L}(\mathbf{x};\theta_0)$ is small, but the cost of drawing samples from the emulator model $\widehat{\mathcal{L}}(\mathbf{x};\theta_0)$ is negligible, then we can instead of a two-sample permutation test perform a goodness-of-fit test, where we draw several independent Monte Carlo (MC) samples of size n_e from $\widehat{\mathcal{L}}(\mathbf{x};\theta_0)$ to produce a set of values $\{\widehat{T}^{(m)}\}_{m=1}^{M}$ that are used as a null distribution to test the hypothesis $\mathcal{L}(\mathbf{x};\theta_0) = \widehat{\mathcal{L}}(\mathbf{x};\theta_0)$. (See Algorithm 11 for details; here $f(\mathbf{x})$ denotes the likelihood $\mathcal{L}(\mathbf{x};\theta_0)$ of the simulator at $\theta = \theta_0$, and $f_e(\mathbf{x})$ denotes the approximate likelihood $\widehat{\mathcal{L}}(\mathbf{x};\theta_0)$ of the emulator at the same parameter value.) If the emulations are cheap, we can choose $n_e \gg n_{\text{sim}}$ as well as a large number M. To cite Friedman (Friedman, 2004, Section IV), the goodness-of-fit approach has "the potential for increased power [compared to two-sample testing] at the expense of having to generate many Monte Carlo samples, instead of just one".

Corollary 4 states that our main result (Theorem 6) still holds for the repeated MC sampling scheme. To simplify the proof, we again use sample splitting for fitting the regression versus computing the test statistic.

Corollary 4. *Suppose that the regression estimator $\widehat{m}(\cdot)$ satisfies*

$$\sup_{m\in\mathcal{M}} \mathbb{E}\int_{\mathcal{X}} (\widehat{m}(\mathbf{x}) - m(\mathbf{x}))^2 dP_X(\mathbf{x}) \leq C_0\delta_n, \tag{3.5}$$

where C_0 is a positive constant, $\delta_n = o(1)$, $\delta_n \geq n^{-1}$ and $\mathcal{M}$ is a class of regression $m(\mathbf{x})$ containing constant functions. Given M such that $\alpha > (M+1)^{-1}$, let us define the test via Monte Carlo sampling by

$$\phi_{\text{MC}} = I\left\{\frac{1}{M+1}\left(1 + \sum_{i=1}^{M} I(\widehat{\mathcal{T}}_{\text{split}}^{(i)} > \widehat{\mathcal{T}}_{\text{split}})\right) \leq \alpha\right\}.$$

Then for fixed $\alpha \in (0,1)$ and $\beta \in (1-\alpha)$ and sufficiently large n_{sim} and n_e, there exists a constant C_1 such that

$$\textit{Type I error: } \mathbb{P}_0(\phi_{\text{MC}} = 1) \leq \alpha,$$

$$\textit{Type II error: } \sup_{m\in\mathcal{M}(C_1\delta_n)} \mathbb{P}_1(\phi_{\text{MC}} = 0) \leq \beta,$$

against the class of alternatives $\mathcal{M}(C_1\delta_n) = \left\{m \in \mathcal{M} : \int_{\mathcal{X}}(m(\mathbf{x}) - \pi_1)^2 dP_X(\mathbf{x}) \geq C_1\delta_n\right\}$.

Note that here, in contrast to the permutation approach, we do not assume that the regression is a linear smoother.

Proof of Corollary 4. We first prove the type I error control and then turn to the type II error control.

- **Type I error.**

With slight abuse of notation, let us write

$$\phi_{\mathrm{MC}}(\mathcal{T}) = I\left\{\frac{1}{M+1}\left(1 + \sum_{i=1}^{M} I(\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)} > \mathcal{T})\right) \leq \alpha\right\},$$

so that $\phi_{\mathrm{MC}}(\widehat{\mathcal{T}}_{\mathrm{split}}) = \phi_{\mathrm{MC}}$. By construction, it can be checked that

$$\frac{1}{M}\sum_{i=1}^{M} \phi_{\mathrm{MC}}(\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)}) \leq \alpha.$$

Furthermore we know that $\widehat{\mathcal{T}}_{\mathrm{split}}$ is equal in distribution to $\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)}$ for any $i = 1, \ldots, M$ under the null hypothesis. Thus

$$\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_0[\phi_{\mathrm{MC}}(\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)})] = \mathbb{E}_0[\phi_{\mathrm{MC}}] \leq \alpha,$$

which verifies the type I error control.

- **Type II error.**

For this part of the proof, we closely follow the proof of Theorem 2.2 in Kim et al. (2019). We let denote the empirical distribution of Monte Carlo samples $\widehat{\mathcal{T}}^{(1)}, \ldots, \widehat{\mathcal{T}}^{(M)}$ by

$$F_M(t) = \frac{1}{M}\sum_{i=1}^{M} I(\widehat{\mathcal{T}}^{(i)} \leq t) \quad \text{for all } t \in \mathbb{R}.$$

Then, by letting $\alpha_M = \alpha(M+1)/M - 1/M$, we can see that $\phi_{\mathrm{MC}} = 1$ if and only if $F_M(t) \geq 1 - \alpha_M$. In other words, we reject the null hypothesis if and only if

$$\widehat{\mathcal{T}}_{\mathrm{split}} \geq c_{1-\alpha_M},$$

where $c_{1-\alpha_M}$ is the upper $1-\alpha_M$ quantile of F_M. One can obtain an upper bound for this quantile by applying Markov's inequality as

$$c_{1-\alpha_M} \leq \frac{1}{\alpha_M}\left(\frac{1}{M}\sum_{i=1}^{M}\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)}\right).$$

Having this observation in mind and putting $\Delta_n = \mathbb{E}[(m(\mathbf{X}) - \pi_1)^2]$, let us define the events $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ such that

$$\begin{aligned}
\mathcal{A}_1 &= \left\{\frac{1}{M}\sum_{i=1}^{M}\widehat{\mathcal{T}}_{\mathrm{split}}^{(i)} \leq 3\beta^{-1}C_0\delta_n\right\}, \\
\mathcal{A}_2 &= \left\{\frac{1}{n}\sum_{i=n+1}^{2n}(\widehat{m}(\mathbf{X}_i) - m(\mathbf{X}_i))^2 \leq 3\beta^{-1}C_0\delta_n\right\} \quad \text{and} \\
\mathcal{A}_3 &= \left\{\left|\frac{1}{n}\sum_{i=n+1}^{2n}(m(\mathbf{X}_i) - \pi_1)^2 - \Delta_n\right| \leq \Delta_n/2\right\}.
\end{aligned}$$

Then applying Markov's inequality together with condition (3.5) yields $\mathbb{P}(\mathcal{A}_1^c) \leq \beta/3$ and $\mathbb{P}(\mathcal{A}_2^c) \leq \beta/3$. Moreover, as shown in Kim et al. (2019), we have $\mathbb{P}(\mathcal{A}_3^c) \leq 4/(C_1 n\delta_n)$. Combining these via the union bound, we see that the type II error is bounded by

$$\begin{aligned}
&\mathbb{P}\left(\widehat{\mathcal{T}}_{\mathrm{split}} < c_{1-\alpha_M}\right) = \mathbb{P}\left(\widehat{\mathcal{T}}_{\mathrm{split}} < c_{1-\alpha_M},\ \mathcal{A}_1\right) + \mathbb{P}\left(\widehat{\mathcal{T}}_{\mathrm{split}} < c_{1-\alpha_M},\ \mathcal{A}_1^c\right) \\
&\leq \mathbb{P}\left(\widehat{\mathcal{T}}_{\mathrm{split}} < 3\alpha_M^{-1}\beta^{-1}C_0\delta_n\right) + \mathbb{P}(\mathcal{A}_1^c) \\
&\leq \mathbb{P}\left(\widehat{\mathcal{T}}_{\mathrm{split}} < 3\alpha_M^{-1}\beta^{-1}C_0\delta_n\right) + \frac{\beta}{3}.
\end{aligned}$$

For the last line, based on the inequality $(x-y)^2 \leq 2(x-z)^2 + 2(z-y)^2$, we further see that

$$\begin{aligned}
&\mathbb{P}\left(\widehat{\mathcal{T}}_{\text{split}} < 3\alpha_M^{-1}\beta^{-1}C_0\delta_n\right) \leq \\
&\quad \leq \ \mathbb{P}\Bigg(\Big(\frac{1}{2n}\sum_{i=n+1}^{2n}(m(\mathbf{X}_i)-\pi_1)^2 - \frac{1}{n}\sum_{i=n+1}^{n}(\widehat{m}(\mathbf{X}_i)-m(\mathbf{X}_i))^2\Big) < 3\alpha_M^{-1}\beta^{-1}C_0\delta_n,\ \mathcal{A}_2\cap\mathcal{A}_3\Bigg) \\
&\qquad\quad + \mathbb{P}\left(\mathcal{A}_2^c \cup \mathcal{A}_3^c\right) \\
&\quad \leq \ \mathbb{P}\left(\Delta_n < 6(1+\alpha_M^{-1})\beta^{-1}C_0\delta_n\right) + \frac{\beta}{3} + \frac{4}{C_1 n\delta_n}.
\end{aligned}$$

Then by taking C_1 sufficiently large, the proof is complete. □

Algorithm 11 Goodness-of-Fit Regression Test via Monte Carlo Sampling

Input: i.i.d. sample $\mathcal{S}$ of size n_{sim} from distribution with density f; emulator model with density f_e; size of Monte Carlo sample n_e; number of additional Monte Carlo samples M; a regression method $\widehat{m}$

Output: p-value for testing if $f(\mathbf{x}) = f_e(\mathbf{x})$ for every $\mathbf{x} \in \mathcal{X}$

```
 1: Let $n = n_{\text{sim}} + n_e$.
 2: Sample $\mathcal{S}_e = \{\mathbf{X}_1^*, \ldots, \mathbf{X}_{n_e}^*\}$ from $f_e$.
 3: Define an augmented sample $\{\mathbf{X}_i, Y_i\}_{i=1}^n$, where $\{\mathbf{X}_i\}_{i=1}^n$=$\mathcal{S} \cup \mathcal{S}_e$, and $Y_i = I(\mathbf{X}_i \in \mathcal{S}_e)$.
 4: Calculate the test statistic $\widehat{\mathcal{T}}$ in Equation 3.4.
 5: for $m \in \{1, \ldots, M\}$ do
 6:     Sample $\mathcal{S}^{(m)} = \{\mathbf{X}_1^{(m)}, \ldots, \mathbf{X}_{n_{\text{sim}}}^{(m)}\}$ from $f$, under the null hypothesis $H_0 : f = f_e$.
 7:     Sample $\mathcal{S}_e^{(m)} = \{\mathbf{X}_1^{*(m)}, \ldots, \mathbf{X}_{n_e}^{*(m)}\}$ from $f_e$.
 8:     Define a new augmented sample $\{\mathbf{X}_i, Y_i\}_{i=1}^n$, where $\{\mathbf{X}_i\}_{i=1}^n$=$\mathcal{S}^{(m)} \cup \mathcal{S}_e^{(m)}$, and $Y_i =$
        $I(\mathbf{X}_i \in \mathcal{S}_e^{(m)})$.
 9:     Refit $\widehat{m}$ and calculate the test statistic on the new augmented sample to obtain $\widehat{\mathcal{T}}^{(m)}$
        from the null distribution $f = f_e$.
10: end for
11: Compute the Monte Carlo $p$-value by $p = \frac{1}{M+1}\left(1 + \sum_{m=1}^M I(\widehat{\mathcal{T}}^{(m)} > \widehat{\mathcal{T}})\right)$.
12: return $p$
```

3.4.2 Approximate P-Values and Confidence Regions

Consider testing $H_0 : \theta \in \Theta_0$. Let $\lambda(\mathbf{x})$ be the likelihood ratio statistic for testing H_0, i.e.,

$$\lambda(\mathbf{x}) = \frac{\sup_{\theta \in \Theta_0} \mathcal{L}(\mathbf{x};\theta)}{\sup_{\theta \in \Theta} \mathcal{L}(\mathbf{x};\theta)}.$$

We estimate $\lambda(\mathbf{x})$ using the estimated likelihood:

$$\widehat{\lambda}(\mathbf{x}) = \frac{\sup_{\theta \in \Theta_0} \widehat{\mathcal{L}}(\mathbf{x};\theta)}{\sup_{\theta \in \Theta} \widehat{\mathcal{L}}(\mathbf{x};\theta)}.$$

The estimated p-value is then

$$\widehat{p}(\mathbf{x}) = \sup_{\theta \in \Theta_0} \mathbb{P}_\theta(\widehat{\lambda}(\mathbf{X}) > \widehat{\lambda}(\mathbf{x}))$$

If $\Theta_0 = \{\theta_0\}$, $\widehat{p}(\mathbf{x})$ can be estimated using data that are simulated under $\theta = \theta_0$. If $|\Theta_0| > 1$, the distribution of the test statistic can be approximated using the χ^2 approximation for the likelihood ratio test (Wilks, 1938). Confidence intervals may be obtained by inverting the hypothesis tests (Neyman, 1937). Overall, as also noted by Cranmer et al. (2020), generating confidence sets by estimating the entire likelihood is less efficient than approaches which estimate the test statistics directly such as the ones presented in Chapter 2 or Brehmer et al. (2020b). Hence, one would expect confidence sets generated by estimating the entire likelihood first to be equally or less powerful than the proposed alternative, keeping the number of simulations the same across the two sets of methods.

Chapter 4

Conditional Density Estimation

Conditional density estimation (CDE) refers to the task of estimating the probability density function $p(y|\mathbf{x})$ of a response variable y given (i.e., conditioned on) features $\mathbf{x}$. Conditional densities are a key component of Bayesian inference, where the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x})$ is the conditional density of the parameter of interest $\boldsymbol{\theta}$ after observing the data $\mathbf{x}$. CDE approaches have been successfully applied in LFI settings for estimating the posterior distribution bypassing the likelihood function (Marin et al., 2016; Papamakarios and Murray, 2016; Lueckmann et al., 2017; Chen and Gutmann, 2019; Izbicki et al., 2019; Greenberg et al., 2019), as well as approximating the likelihood itself (Papamakarios et al., 2019; Lueckmann et al., 2019; Wiqvist et al., 2021). CDE methods are also suitable for prediction settings where heteroskedastic errors or multimodal response may occur, and hence accounting for the full prediction uncertainty in the response y becomes necessary in any downstream task. For example, in precision cosmology one needs to combine data from different scientific probes, each affected by unique sources of systematic uncertainty, to produce samples from complicated joint likelihood functions with nontrivial covariances in a high-dimensional parameter space (Krause et al., 2017; Joudaki et al., 2017; Aghanim et al., 2020; van Uitert et al., 2018; Abbott et al., 2019). In such situations, CDE methods that target a variety of settings and non-standard data (images, correlation functions, mixed data types) become especially valuable. However, for any given data type, there is no one-size-fits-all CDE method. For example, deep neural networks often perform well in settings with large amounts of representative training data but in applications with smaller training samples one may need a different tool. There is also additional value in models that are interpretable and easy to fit to the data at hand. In this chapter we follow Izbicki et al. (2014, 2019) to combine ABC and CDE by directly applying CDE techniques and loss functions (Section 4.2.1) to simulated data. We first review a set of tools which leverage a large range of machine learning algorithms for conditional density estimation: `NNKCDE`, `RFCDE`/`fRFCDE`, which use a neighbor-based approach, and `FlexCode`, which approximates the posterior distribution using an orthogonal series expansion. We also introduce `DeepCDE`, a CDE tool which extends `FlexCode` to virtually any neural network architecture for regression and can be extend to non-standard data such as images and time series. We conclude by showcasing three applications of `DeepCDE` (i) wildfire size and duration prediction for the continental United States, (ii) predicting orientation of simulated galaxy images using the `GalSim` toolkit (Rowe et al., 2015) and (iii) prediction benzene concentration in the air using time series measurements from chemical sensors.

4.1 Review: A Suite of Tools for Conditional Density Estimation

We start by briefly describing the conditional density estimators in Tables 4.1 and 4.2. Unless otherwise stated, we choose the tuning or hyper-parameters by minimizing the CDE empirical loss in Equation 4.4 using cross-validation.

4.1.1 `NNKCDE`

Nearest-Neighbors Kernel CDE (NNKCDE; Izbicki et al. 2017, Freeman et al. 2017) is a simple and easily interpretable CDE method. It computes a kernel density estimate of $\mathbf{y}$ using the k nearest neighbors of the evaluation point $\mathbf{x}$. The model has only two tuning parameters: the number of nearest neighbors k and the bandwidth h of the smoothing kernel in $\mathbf{y}$-space. Both tuning parameters are chosen in a principled way by minimizing the CDE loss on validation data.

Method	**Name**	**Summary**	**Hyper-parameters**
`NNKCDE`	`N`earest `N`eighbor `K`ernel `CDE`	Computes a KDE estimate of multivariate $\mathbf{y}$ using the nearest neighbors of the evaluation point $\mathbf{x}$ in feature space.	• Number of neighbors k • Kernel bandwidth h
`RFCDE`	`R`andom `F`orest `CDE`	Random forest that partitions the feature space using a CDE loss. Constructs a weighted KDE estimate of multivariate $\mathbf{y}$ with weights defined by leaves in the forest.	• Random forest hyperparams. • Kernel bandwidth h
`fRFCDE`	`f`unctional `R`andom `F`orest `CDE`	`RFCDE` version suitable for functional features $\mathbf{x}$. Partitions the feature space directly rather than representing $\mathbf{x}$ as a vector.	• Random forest hyperparams. • Kernel bandwidth h • Partition parameter λ
`FlexCode`	`Flex`ible `Co`nditional `D`ensity `E`stimation	Uses basis expansion of univariate y to turn CDE into a series of univariate regression problems.	• Number of expansion coeffs. • Selected regression method hyperparams.
`DeepCDE`	`Deep` Neural Networks `CDE`	Uses basis expansion of univariate y similar to `FlexCode`, but learns the expansion coefficients simultaneously using a deep neural network.	• Number of expansion coeffs. • Selected deep neural network architecture hyperparams.

Table 4.1: Naming convention, high-level summary and hyper-parameters of CDE methods, along with references for further details and code examples.

Method	Capacity (# Training Pts)	Multivariate Response	Functional Features	Image Features
`NNKCDE`	Up to $\sim 10^5$	✓		
(f)`RFCDE`	Up to $\sim 10^6$	✓	✓	
`FlexCode`	Up to $\sim 10^6$		✓	
`DeepCDE`	Up to $\sim 10^8$		✓	✓

Table 4.2: Comparison of CDE methods in terms of training capacity and compatibility with multivariate response and different types of features, in descending order of capacity. Capacities are estimated based on input with around 100 features and a standard i5/i7/quad-core processor with 16GB of RAM. Note that less complex methods (such as `NNKCDE`) tend to be easier to use, easier to interpret, and often perform better in settings with smaller training sets, whereas more complex methods (such as `DeepCDE`) perform better in settings with larger (representative) training sets.

More specifically, the kernel density estimate of $\mathbf{y}$ given $\mathbf{x}$ is defined as

$$\widehat{p}(\mathbf{y}|\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} K_h\left[\rho(\mathbf{y}, \mathbf{y}_{s_i(\mathbf{x})})\right], \tag{4.1}$$

where K_h is a normalized kernel (e.g., a Gaussian function) with bandwidth h, ρ is a distance metric, and $s_i(\mathbf{x})$ is the index of the i^{th} nearest neighbor of $\mathbf{x}$. It is essentially a smoother version of the histogram estimator proposed by Cunha et al. (2009) in that it approximates the density with a smooth continuous function rather than by binning.

4.1.2 `RFCDE` and `fRFCDE`

Random forests (RFs, Breiman 2001) is one of the best off-the-shelf solutions for regression and classification problems. It builds a large collection of decorrelated trees, where each tree is a data-based partition of the feature space. The trees are then averaged to yield a prediction. RFCDE, introduced by Pospisil and Lee (2018), is an extension of random forests to conditional density estimation and multivariate responses. Like `NNKCDE`, it computes a kernel density estimate of $\mathbf{y}$ but with nearest neighbor weightings defined by the location of the evaluation point $\mathbf{x}$ relative to the leaves in the random forest. RFCDE inherits the advantages of random forests in that it can handle mixed-typed data. It also does not require the user to specify distances or similarities between data points, and it has good performance while remaining relatively interpretable.

The main departure from other random forest algorithms is our criterion for feature space partitioning decisions. In regression contexts, the splitting variable and split point are typically chosen so as to minimize the mean-squared-error loss. In classification contexts, the splits are typically chosen so as to minimize a classification error. Existing random forest density estimation methods such as quantile regression forests by Meinshausen (2006) and the TPZ algorithm by Carrasco Kind and Brunner (2013) use the same tree structure as regression and classification random forests, respectively. `RFCDE`, however, builds trees that minimize the CDE loss (see Equation 4.4), allowing the forest to adapt to structures in the conditional density; hence overcoming some of the limitations of the usual regression approach for data with heteroskedasticity and multimodality. In addition, `RFCDE` does not require discretizing the response as in TPZ, thereby providing more accurate results at a lower cost for continuous responses, especially in the case of multivariate continuous responses where binning is problematic. See Pospisil and Lee (2018) for further examples and comparisons.

Another unique feature of `RFCDE` is that it can handle multivariate responses with joint densities by applying a weighted kernel smoother to $\mathbf{y}$. This added feature enables analysis of complex conditional distributions that describe relationships between multiple responses and features, or equivalently between multiple parameters and observables in an LFI setting. Like quantile regression forests, the `RFCDE` algorithm takes advantage of the fact that random forests can be viewed as a form of adaptive nearest-neighbor method with the aggregated tree structures determining a weighting scheme. This weighting scheme can then be used to estimate the conditional density $p(\mathbf{y}|\mathbf{x})$, as well as the conditional mean and quantiles, as in quantile regression forests (but for CDE-optimized trees). As mentioned above, `RFCDE` computes the latter density by a weighted kernel density estimate (KDE) in $\mathbf{y}$ using training points near the evaluation point $\mathbf{x}$. These distances are effectively defined by how often a training point $\mathbf{x}_i$ belongs to the same leaf node as $\mathbf{x}$ in the forest (see Equation 1 in Pospisil and Lee 2018 for details).

Despite the increased complexity of our CDE trees, `RFCDE` still scales to large data sets because of an efficient computation of splits via orthogonal series. Moreover, `RFCDE` extends the density estimates on new $\mathbf{x}$ to the multivariate case through the use of multivariate kernel density estimators (Epanechnikov, 1969). In both the univariate and multivariate cases, bandwidth selection can be handled by either plug-in estimators or by tuning using a density estimation loss.

`fRFCDE` (Pospisil and Lee, 2019) is a variant of `RFCDE` that can accommodate functional features $\mathbf{x}$ by partitioning in the continuous domain of such features. The spectral energy distribution (SED) of a galaxy is its energy as a function of continuous wavelength λ of light; hence it can be viewed as functional data. Another example of functional data is the shear correlation function of weak lensing, which measures the mean product of the shear at two points as a function of a continuous distance r between those points. Similarly, any function of continuous time is an example of functional data. Treating functional features (like spectra, correlation functions or images) as unordered multivariate vectors on a grid suffers from a curse of dimensionality. As the resolution of the grid becomes finer the dimensionality of the data increases but little additional information is added, due to high correlation between nearby grid points. `fRFCDE` adapts to this setting by partitioning the domain of each functional feature (or curve) into intervals, and passing the mean values of the function in each interval as input to `RFCDE`. Feature selection is then effectively done over regions of the domain rather than over single points. More specifically, the partitioning in `fRFCDE` is governed by the parameter μ of a Poisson process, with each functional feature entering as a high-dimensional vector $\mathbf{x} = (x_1, \ldots, x_d)$. Starting with the first element of the vector, we group the first Poisson(μ) elements together. We then repeat the procedure sequentially until we have assigned all d elements into a group; this effectively partitions the function domain into disjoint intervals $\{(l_i, h_i)\}$. The function mean values or smoothed brightness measurements $\widetilde{x}_i \equiv \int_{l_i}^{h_i} f(\lambda) d\lambda$ of each interval are finally treated as new inputs to a standard (vectorial) `RFCDE` tree. The splitting of the smoothed predictors $\widetilde{x}_i$ is done independently for each tree in the forest. Other steps of `fRFCDE`, such as the computation of variable importance, also proceed as in (vectorial) `RFCDE` but with the averaged values of a region as inputs. As a result, `fRFCDE` has the capability of identifying the functional inputs and the regions in the input domain that are important for estimating the response $\mathbf{y}$. Figure 4.1 shows schematically the differences and similarities in construction between standard `RFCDE` and its `fRFCDE` variant.

4.1.3 `FlexCode`

Introduced by Izbicki and Lee (2017), `FlexCode`is a CDE method that uses a basis expansion for the univariate response y and poses CDE as a series of univariate regression problems. The main advantage of this method is its flexibility as any regression method can be applied towards CDE, enabling us to tailor our choice of regression method to the intrinsic structure and type of data at hand.

More precisely, let $\{\phi_j(y)\}_j$ be an orthonormal basis like a Fourier or wavelet basis for functions of $y \in \mathbb{R}$. The key idea of `FlexCode` is to express the unknown density $p(y|\mathbf{x})$ as a basis expansion

$$p(y|\mathbf{x}) = \sum_j \beta_j(\mathbf{x}) \phi_j(y). \tag{4.2}$$

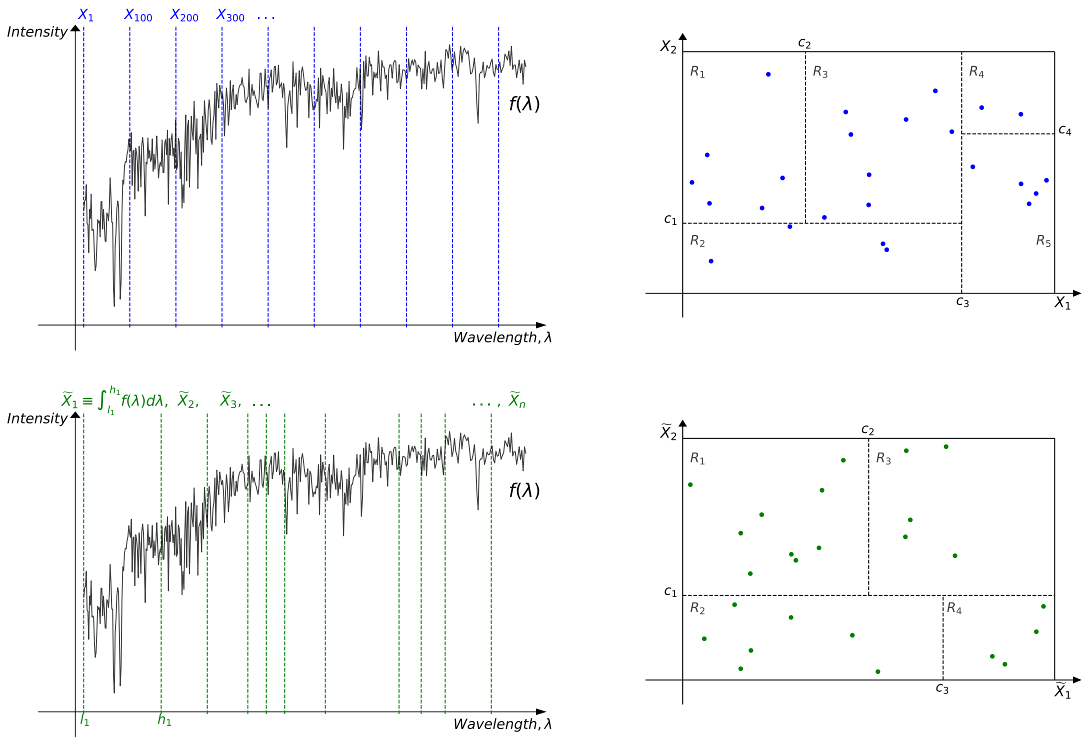


Figure 4.1: A schematic diagram of RFCDE (top row) and fRFCDE (bottom row) applied to a galaxy spectrum from Vanderplas et al. (2012). *Top row*: RFCDE treats the intensity x_i at each recorded wavelength λ_i of the spectrum as a feature or "input" to the random forests algorithm — the blue vertical dashed lines indicate every 100th recorded wavelength. RFCDE then builds an ensemble of CDE trees, where each tree partitions the feature space according to the CDE loss, as illustrated in the top right figure for features x_1 and x_2. *Bottom row*: fRFCDE instead groups nearby measurements together where the group divisions are defined by a Poisson process with parameter μ (vertical green dashed lines, left figure). The new smoothed features $\widetilde{x}_i$ are computed by integrating the intensity over the grouped wavelengths. A forest of CDE trees is then built using the same construction as in RFCDE but with the smoothed features as inputs (bottom right figure).

By the orthogonality property of the basis, the (unknown) expansion coefficients $\{\beta_j(\mathbf{x})\}_j$ are then just orthogonal projections of $p(y|\mathbf{x})$ onto the basis vectors. We can estimate these coefficients using a training set of $(\mathbf{x}, y)$ data by *regressing* the transformed response variables $\phi_j(y)$ on predictors $\mathbf{x}$ for every basis function j (see Izbicki and Lee (2017) Equation 2.2, for details). The number of basis function n_{basis} is chosen by minimizing a CDE loss function on validation data. The estimated density, $\sum_{j=1}^{n_{basis}} \widehat{\beta}_j(\mathbf{x})\phi_j(y)$, may contain small spurious bumps induced by the Fourier approximation and may not integrate to one. We remove such artifacts as described in Izbicki and Lee (2016) by applying a thresholding parameter δ chosen via cross-validation. `FlexCode` turns a challenging density estimation problem into a simpler regression problem, where we can choose any regression method that fits the problem at hand.

To provide a concrete application example, Schmidt et al. (2020) present the results of an initial study of the LSST Dark Energy Science Collaboration (`LSST-DESC`) for photometric redshift estimation or "photo-z". In photo-$z$estimation, one attempts to constrain the cosmological *redshift* (z) of a galaxy after observing the shifted spectrum using a handful of broadband filters, and sometimes additional variables such as morphology and environment. Their initial data challenge ("Photo-z DC 1") compares the CDEs of a dozen photo-z codes run on simulations of LSST galaxy photometry catalogs in the presence of complete, correct, and representative training data. `FlexZBoost`, a version of `FlexCode` based on the scalable gradient boosting regression technique by Chen and Guestrin (2016), was entered into the data challenge because of the method's ability to scale to massive data. In the DC1 analysis, `FlexZBoost` was among the the strongest performing codes according to established performance metrics of such PDFs and was one of only two codes to beat the experimental control under a more discriminating metric, the CDE loss. For massive surveys such as LSST, `FlexCode` also has another advantage compared to other CDE methods, namely its compact, lossless storage format. Juric et al. (2017) establishes that LSST has allocated $\sim$100 floating point numbers to quantify the redshift of each galaxy. As is shown in Schmidt et al. (2020), the myriad methods for deriving photo-z PDFs yield radically different results, motivating a desire to store the results of more than one algorithm in the absence of an obvious best choice. For the photo-z PDFs of most codes, one may need to seek a clever storage parameterization to meet LSST's constraints (Carrasco Kind and

Brunner, 2014; Malz et al., 2018), but `FlexCode` is virtually immune to this restriction. Since `FlexCode` relies on a basis expansion, one only needs to store n_{basis} coefficients per target density for a lossless compression of the estimated PDF with no need for binning. Indeed, for DC1, we can with `FlexZBoost` reconstruct our estimate $\widehat{p}(z|\mathbf{x})$ at any resolution from estimates of the first 35 coefficients in a Fourier basis expansion. In other words, `FlexZBoost` enables the creation and storage of high-resolution photo-z catalogs for several billion galaxies at no added cost.

4.2 Review: Model Selection and Assessment

After fitting CDEs, it is important to assess the quality of our models of uncertainty. In the LFI task, a key question is whether an estimate of the posterior distribution, $\widehat{p}(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})$ of the cosmological parameters is close enough to the true posterior $p(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})$ given the observations $\mathbf{x}_{\text{obs}}$. First, we describe a CDE loss function that directly provides relative comparisons between conditional density estimators or, equivalently, between a set of models (for the same method) with different tuning parameters. Second, we describe visual diagnostic tools, such as Probability Integral Transforms (PIT) and Highest Probability Density (HPD) plots, that can provide insights on the overall goodness-of-fit of a given estimator to observed data.

4.2.1 CDE loss

Here we briefly review the CDE loss from Izbicki and Lee (2016) for assessing conditional density estimators and discuss it in the context of the cosmology LFI case.

The goal of a loss function is to provide relative comparisons between different estimators, so that it is easy to directly choose the best fitted model among a list of candidates. Given an estimate $\widehat{p}$ of p, we define the CDE loss by

$$L(\widehat{p}, p) = \int\int \left[\widehat{p}(\mathbf{y}|\mathbf{x}) - p(\mathbf{y}|\mathbf{x})\right]^2 d\mathbf{y} dP(\mathbf{x}), \tag{4.3}$$

where $P(\mathbf{x})$ is the marginal distribution of the features $\mathbf{x}$. This loss is the CDE analog to the standard mean squared error (MSE) in standard regression. The weighting by the marginal distribution of the features emphasizes that errors in the estimation of $\mathbf{y}$ for unlikely features $\mathbf{x}$ are less important. The CDE loss cannot be directly evaluated because it depends on the unknown true density $p(z|\mathbf{x})$. However, one can estimate the loss (up to a constant determined by the true p) by

$$\widehat{L}(\widehat{p}, p) = \frac{1}{n} \sum_{i=1}^{n} \int \widehat{p}(\mathbf{y}|\mathbf{x}_i^{te})^2 d\mathbf{y} - \frac{2}{n} \sum_{i=1}^{n} \widehat{p}(\mathbf{y}_i^{te}|\mathbf{x}_i^{te}), \tag{4.4}$$

where $\left\{(\mathbf{x}_i^{te}, \mathbf{y}_i^{te})\right\}_{i=1}^{n}$ represents our validation or test data, i.e., a held-out set not used to construct $\widehat{p}$.

CDE loss for LFI. In LFI settings, we use a slightly different version of the CDE loss in Eq. 4.3. Because the goal (in ABC) is to approximate the posterior density $p(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})$ at fixed $\mathbf{x} = \mathbf{x}_{\text{obs}}$, a natural evaluation metric is the integrated squared error loss

$$\int \left[\widehat{p}(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}}) - p(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})\right]^2 d\boldsymbol{\theta} \tag{4.5}$$

of the conditional density *at* $\mathbf{x}_{obs}$ *only*. Estimating this loss can however be tricky as only a single instance of data with $x = \mathbf{x}_{\text{obs}}$ is available in practice. Hence, Izbicki et al. (2019) approximates Equation 4.5 by computing the empirical loss $\widehat{L}(\widehat{p}, p)$ in Eq. 4.4 over a restricted subset of the validation data that only includes the $\mathbf{x}_i^{te}$ points that fall in an ϵ-neighborhood of $\mathbf{x}_{\text{obs}}$, where ϵ is the tolerance of the ABC rejection algorithm. The detailed analysis of this approach can be found in Izbicki et al. (2019).

4.2.2 PIT and HPD diagnostics

The CDE loss function is a relative measure of performance that cannot address absolute goodness-of-fit. To quantify overall goodness-of-fit, we examine how well calibrated an ensemble of conditional density estimators are on average, over validation or test data $\left\{(\mathbf{x}_i^{te}, \mathbf{y}_i^{te})\right\}_{i=1}^{n}$. For ease of notation, we will in this section denote $\mathbf{x}_i^{te}$ and $\mathbf{y}_i^{te}$ for a generic i by $\mathbf{x}_{val}$ and $\mathbf{y}_{val}$.

Given a true probability density $p(\mathbf{y}|\mathbf{x}) = \gamma_0$ of a variable $\mathbf{y}$ conditioned on data $\mathbf{x}$, an estimated probability density $\widehat{p}(\mathbf{y}|\mathbf{x}) = \gamma$ cannot be *well-calibrated* unless $\gamma \approx \gamma_0$. Built on the same logic, the probability integral transform (PIT; D'Isanto 2016)

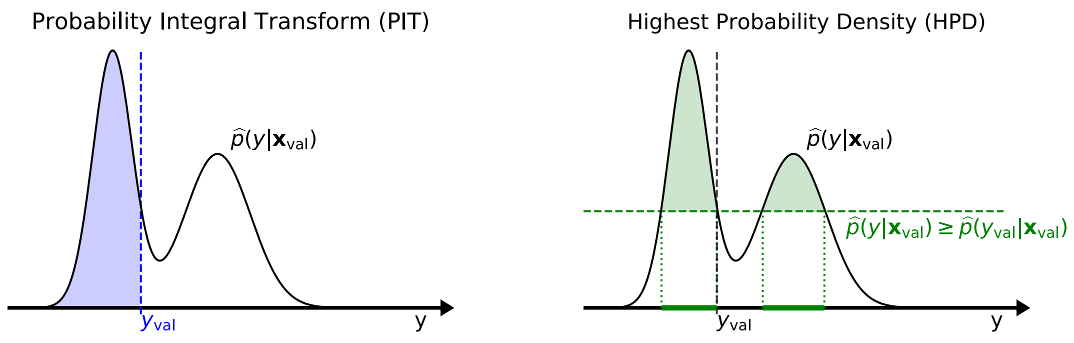
$$PIT(\mathbf{x}_{\text{val}}, y_{\text{val}}) = \int_{-\infty}^{y_{\text{val}}} \widehat{p}(y|\mathbf{x}_{\text{val}})dy \tag{4.6}$$

assesses the calibration quality of an ensemble of CDEs for scalar y representing the cumulative distribution function (CDF) of $\widehat{p}(y|\mathbf{x}_{\text{val}})$ evaluated at $y = y_{\text{val}}$; this PIT value corresponds to the shaded area in Figure 4.2, left. A statistically self-consistent population of densities has a uniform distribution of PIT values, and deviations from uniformity indicate inaccuracies of the estimated PDFs. Overly broad CDEs manifest as under-representation of the lowest and highest PIT values, whereas overly narrow CDEs manifest as over-representation of the lowest and highest values.

However, PIT values do not easily generalize to multiple responses. For instance, for a bivariate response $\mathbf{y} = (z, \eta)$, the quantity $\int_{-\infty}^{z_{\text{val}}} \int_{-\infty}^{\eta_{\text{val}}} p(z, \eta|\mathbf{x}_{\text{val}})dzd\eta$ is not in general uniformly distributed (Genest and Rivest, 2001). An alternative statistic that easily generalizes to multivariate $\mathbf{y}$ is the highest probability density value (HPD; Izbicki et al. 2017, Appendix A):

$$\xi(\mathbf{x}_{\text{val}}, \mathbf{y}_{\text{val}}) = \int_{\mathbf{y}:\widehat{p}(\mathbf{y}|\mathbf{x}_{\text{val}}) \geq \widehat{p}(\mathbf{y}_{\text{val}}|\mathbf{x}_{\text{val}})} \widehat{p}(\mathbf{y}|\mathbf{x}_{\text{val}})d\mathbf{y}. \tag{4.7}$$

The HPD value is based on the definition of the *highest density region* (HDR, Hyndman 1996) of a random variable $\mathbf{y}$; that is, the subset of the sample space of $\mathbf{y}$ where all points in the region have a probability above a certain value. The HDR of $\mathbf{y}|\mathbf{x}_{\text{val}}$ can be seen as a region estimate of $\mathbf{y}$ when $\mathbf{x} = \mathbf{x}_{\text{val}}$ is observed. In words, the set $\{\mathbf{y} : \widehat{p}(\mathbf{y}|\mathbf{x}_{\text{val}}) \geq \widehat{p}(\mathbf{y}_{\text{val}}|\mathbf{x}_{\text{val}})\}$ is the smallest HDR containing the point $\mathbf{y}_{\text{val}}$, and the HPD value is simply the probability of such a region. Figure 4.2, right, shows a schematic diagram of the HPD value (green shaded area) and HDR region (highlighted segments on the y-axis) for the estimated density $\widehat{p}(\mathbf{y}|\mathbf{x}_{\text{val}})$. The HPD value $\xi(\mathbf{x}_{\text{obs}}, \mathbf{y}_{\text{val}})$ can also be viewed as a measure of how plausible $\mathbf{y}_{\text{val}}$ is according to $\widehat{p}(\mathbf{y}|\mathbf{x}_{\text{val}})$ and is directly related to the Bayesian analog of p-values or *the e-value* (Pereira and Stern, 1999). One can show (Harrison et al., 2015) that even for multivariate $\mathbf{y}$, the HPD values for validation data follow a $U(0, 1)$ distribution if the CDEs are well-calibrated on the population level. Thus, these values can also be used



Probability Integral Transform (PIT)
$\widehat{p}(y|\mathbf{x}_{\mathrm{val}})$
y_{val}
y
Highest Probability Density (HPD)
$\widehat{p}(y|\mathbf{x}_{\mathrm{val}})$
$\widehat{p}(y|\mathbf{x}_{\mathrm{val}}) \geq \widehat{p}(y_{\mathrm{val}}|\mathbf{x}_{\mathrm{val}})$
y_{val}
y

Figure 4.2: Schematic diagram of the construction of the Probability Integral Transform (PIT, left) and the Highest Probability Density (HPD, right) values for the estimated density $\widehat{p}(y|\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_{\mathrm{val}}$, where y_{val} is the response at $x = \mathbf{x}_{\mathrm{val}}$. In the plot to the right, the highlighted segments on the y-axis form the so-called highest density region (HDR) of $y|\mathbf{x}_{\mathrm{val}}$. The PIT and HPD values correspond to the area of the tail versus highest density region, respectively, of the estimate; here indicated by the blue versus green shaded areas.

for assessing the fit of conditional densities in the same way as PIT values. Additionally, PIT and HPD values can be used for validating the coverage of conditional density models at any level $\alpha \in [0, 1]$ (Zhao et al., 2021).

The PIT and HPD are not without their limitations, however, as demonstrated in the control case of Schmidt et al. (2020). Because the PIT and HPD values can be uniformly distributed even if $p(\mathbf{y}|\mathbf{x})$ is not well estimated, such as when using as conditional model the marginal distribution of the data, they must be used in conjunction with loss functions for method assessment. A popular way of visualizing PIT and HPD diagnostics for the entire population is through *probability-probability plots* or P-P plots of the empirical distribution of the (PIT or HPD) statistic versus its distribution under the hypothesis that $\widehat{p}(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})$; henceforth, we will refer to the latter Uniform(0,1) distribution as the "theoretical" distribution of PIT or HPD. An ideal P-P plot has all points close to the identity line where the "empirical" and "theoretical" distributions are the same. Note that HPD P-P plots, in particular, are valuable calibration tools if our goal is to calibrate the estimated densities so that the computed predictive regions have the right coverage.

4.3 DeepCDE: Leveraging Deep Neural Networks for Conditional Density Estimation

Recently, neural networks have reemerged as a powerful tool for prediction settings where large amounts of representative training data are available; see LeCun et al. (2015) and Goodfellow et al. (2016) for a full review. Neural networks for CDE, such as Mixture Density Networks (MDNs; Bishop 1994) and variational methods (Tang and Salakhutdinov, 2013; Sohn et al., 2015), usually assume a Gaussian or some other parametric form of the conditional density. MDNs have lately also been used for photometric redshift estimation (D'Isanto and Polsterer, 2018; Pasquet et al., 2019) and for direct estimation of likelihoods and posteriors in cosmological parameter inference (see Alsing et al. 2019 and references within).

DeepCDEtakes a different, *fully nonparametric* approach to CDE. It combines the advantages of basis expansions with the flexibility of neural network architectures, allowing for data types like image features and time-series data. DeepCDE can be implemented with both convolutional and recurrent neural network architectures, extending to both image and sequential data; we showcase this with in Sections 4.4 and 4.4 respectively. DeepCDE is based on the orthogonal series representation in FlexCode, given in Equation 4.2, but rather than relying on regression methods to estimate the expansion coefficients in Equation 4.2, DeepCDE computes the coefficients $\{\beta_i(\mathbf{x})\}_{i=1}^{B}$ *jointly* with a neural network that minimizes the CDE loss in Equation 4.3. Indeed, for an orthogonal basis the problem of minimizing this CDE loss is (asymptotically) equivalent to finding the best basis coefficients in FlexCode under mean squared error loss for the individual regressions, as shown by the following Lemma.

Lemma 2. *Let $\boldsymbol{\beta} = \{\beta_i(\mathbf{x})\}_{i=1}^{B}$ be the coefficients of the FlexCode basis expansion in Equation 4.2. Minimizing the CDE loss in Equation 4.3 is equivalent to minimizing the mean squared errors of the basis expansion coefficients, i.e.,*

$$\min_{\widehat{\boldsymbol{\beta}} \in \mathbb{R}^B} \int_{\mathcal{X}} \int_{\mathcal{Y}} (\widehat{p}(y|\mathbf{x}) - p(y|\mathbf{x}))^2 dy dP(\mathbf{x}) \Longleftrightarrow \min_{\widehat{\boldsymbol{\beta}} \in \mathbb{R}^B} \mathbb{E}_{\mathbf{x}} \left[\left\| \widehat{\boldsymbol{\beta}}(\mathbf{x}) - \boldsymbol{\beta}(\mathbf{x}) \right\|^2 \right]. \tag{4.8}$$

Proof. We prove the statement by showing that the two minimization problems are equivalent.

First, considering the LHS of Equation 4.8, we have that:

$$
\begin{aligned}
&\min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \int_{\mathcal{X}}\int_{\mathcal{Y}} (\widehat{p}(y|\mathbf{x}) - p(y|\mathbf{x}))^2 dy dP(\mathbf{x}) \\
&\Longleftrightarrow \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \int_{\mathcal{X}}\int_{\mathcal{Y}} \widehat{p}(y|\mathbf{x})^2 dy dP(\mathbf{x}) - 2\int_{\mathcal{X}}\int_{\mathcal{Y}} \widehat{p}(y|\mathbf{x}) p(\mathbf{x},y) d\mathbf{x} dy \\
&= \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \int_{\mathcal{X}}\int_{\mathcal{Y}} \left(\sum_{i=1}^{B} \widehat{\beta}_i(\mathbf{x})\phi_i(y)\right)^2 dy dP(\mathbf{x}) - 2\int_{\mathcal{X}}\int_{\mathcal{Y}} \left(\sum_{i=1}^{B} \widehat{\beta}_i(\mathbf{x})\phi_i(y)\right) p(y|\mathbf{x}) dP(\mathbf{x}) dy \\
&= \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \int_{\mathcal{X}} \sum_{i,j=1}^{B} \widehat{\beta}_i(\mathbf{x})\widehat{\beta}_j(\mathbf{x}) \int_{\mathcal{Y}} \phi_i(y)\phi_j(y) dy dP(\mathbf{x}) \\
&\qquad - 2\int_{\mathcal{X}}\int_{\mathcal{Y}} \left(\sum_{i=1}^{B} \widehat{\beta}_i(\mathbf{x})\phi_i(y)\right)\left(\sum_{j=1}^{B} \beta_j(\mathbf{x})\phi_j(y)\right) dP(\mathbf{x}) dy \\
&= \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \int_{\mathcal{X}} \sum_{i=1}^{B} \widehat{\beta}_i^2(\mathbf{x}) dP(\mathbf{x}) - 2\int_{\mathcal{X}} \sum_{i=1}^{B} \widehat{\beta}_i(\mathbf{x})\beta_i(\mathbf{x}) dP(\mathbf{x}) = \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \sum_{i=1}^{B} \mathbb{E}_{\mathbf{x}}\left[\widehat{\beta}_i^2(\mathbf{x}) - 2\widehat{\beta}_i(\mathbf{x})\beta_i(\mathbf{x})\right],
\end{aligned}
$$

where the second equality follows from the fact that the set $\{\phi_i(y)\}_{i=1}^{B}$ is part of an orthonormal basis, that is,

$$
\int_{\mathcal{Y}} \phi_i(y)\phi_j(y) dy = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.
$$

Next, the RHS of Equation 4.8 reduces to:

$$
\begin{aligned}
\min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \mathbb{E}_{\mathbf{x}}\left[\left\|\widehat{\boldsymbol{\beta}}(\mathbf{x}) - \boldsymbol{\beta}(\mathbf{x})\right\|^2\right] &= \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \sum_{i=1}^{B} \mathbb{E}_{\mathbf{x}}\left[\left(\widehat{\beta}_i(\mathbf{x}) - \beta_i(\mathbf{x})\right)^2\right] \\
&\Longleftrightarrow \min_{\widehat{\boldsymbol{\beta}}\in\mathbb{R}^B} \sum_{i=1}^{B} \mathbb{E}_{\mathbf{x}}\left[\widehat{\beta}_i^2(\mathbf{x}) - 2\widehat{\beta}_i(\mathbf{x})\beta_i(\mathbf{x}).\right]
\end{aligned}
$$

□

The value of this result is that DeepCDE with a CDE loss directly connects prediction with uncertainty quantification, implying that one can leverage the state-of-the-art deep architectures for an application at hand toward uncertainty quantification for the same prediction setting. In addition, DeepCDE retains the same benefits as FlexCode in regards to storage space, that is one would only need to store the basis expansion coefficients for a lossless compression of the conditional density estimate.

From a neural network architecture perspective, DeepCDE only adds a linear output layer of coefficients for a series expansion of the density according to

$$\widehat{p}(y|\mathbf{x}) = \sum_{j=1}^{B} \widehat{\beta}_j(\mathbf{x})\phi_j(y), \tag{4.9}$$

where $\{\phi_j(y)\}_{j=1}^{B}$ is an orthogonal basis for functions of $y \in \mathbb{R}$. Like FlexCode, we normalize and remove spurious bumps from the final density estimates according to the procedure in Section 2.2 of Izbicki and Lee (2016). For most deep architectures, adding a linear layer represents a small modification, and a negligible increase in the number of parameters. For instance, with the AlexNet architecture (Krizhevsky et al., 2012), a widely used, relatively shallow convolutional neural network, adding a final layer with 30 coefficients for a cosine basis only adds $\sim 120,000$ extra parameters. This represents a 0.1% increase over the 62 million already existing parameters, and hence a negligible increase in training and prediction time. Moreover, the CDE loss for DeepCDE is especially easy to evaluate, as inserting the orthogonal basis expansion (Equation 4.9) into the CDE loss (Equation 4.3) yields

$$\begin{aligned} L(p,\widehat{p}) &= \int \sum_{j=1}^{B} \widehat{\beta}_j^2(\mathbf{x})dP(\mathbf{x}) - 2\int\int \sum_{j=1}^{B} \widehat{\beta}_j(\mathbf{x})\phi_j(y)dP(\mathbf{x},y) \\ &\approx \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{B}\widehat{\beta}_j^2(\mathbf{x}_i^{te}) - 2\sum_{j=1}^{B}\widehat{\beta}_j(\mathbf{x}_i^{te})\phi_j(y_i^{te})\right). \end{aligned} \tag{4.10}$$

The last expression represents our empirical CDE loss on validation data $\left\{(\mathbf{x}_i^{te}, y_i^{te})\right\}_{i=1}^{n}$ and is easy to compute.

4.4 Application: Correcting Galaxy Image Orientation

In this section we provide an example of estimating conditional density estimates when the input is an image, which is a high-dimensional structured input. As in Section 2.9.5, we use open-source galaxy simulator `GALSIM` (Rowe et al., 2015), which allows to generated realistic images of astronomical objects and integrates real observational effects such as pixelization and blurring. For this simulation experiment we only consider symmetrical globular galaxies, that is a spherical group of stars rotating around the galaxy center, with an axis ratio equal to $\lambda = 0.4$. We down-sample the simulated galaxy images to a 20×20 resolution to mimic realistic observations as in Izbicki et al. (2014).

We consider two different settings. In setting (A), we randomly select the simulation galaxy orientation θ from the interval $[0, \pi]$ (in radians), or $[0°, 180°]$ (in degrees). In setting (B), we select θ from the inteval $[-\pi, \pi]$ (in radians), or $[-180°, 180°]$ (in degrees). In both settings we want to correct the galaxy image rotation, that is to estimate the conditional density estimate of the galaxy orientation θ given the 400-dimensional galaxy image as input. In setting (A) the rotation angle is unique as we are restricting the angle from $[0, \pi]$. In this case, the conditional density estimate should be a peaked distribution around the true rotation angle. In setting (B), however, given any rotation angle $\widetilde{\theta}$ the angle $(\widetilde{\theta} - \pi)$ would also be a valid rotation angle. In setting (B) the conditional density estimate would identify this correspondence, showing a bi-modal distribution about the two potential values. Figure 4.3 provides three examples of a simulated galaxy rotation, with the first two figures showing the correspondence of setting (B): there is no difference in the final image if a galaxy is rotated by $16°$ or by $-165°$. In both settings we employ four different deep neural network architectures: (i) a 3-layer multi-layer perceptron (MLP) or feed-forward neural network, (ii) a simplified version of the AlexNet model (Krizhevsky et al., 2012), with two convolutional layers and two linear layers, and two residual networks (ResNets, He et al. 2016) with (iii) 18 layers (Resnet18) and (iv) 34 layers (Resnet34). We compare the performance of all models against the marginal distribution, which passes the PIT and HPD diagnostics as mentioned in Section 4.2.2, but does not represent a good model for the conditional density. We explore the use of the Fourier cosine basis functions and the Haar wavelet basis functions (Haar, 1910), both orthonormal basis functions. Finally, since in this specific the correct conditional densities are either a single peak or two-peaked distribution, we will not consider HPD diagnostics, given that in the majority of the cases the highest posterior density region would have zero mass by design.

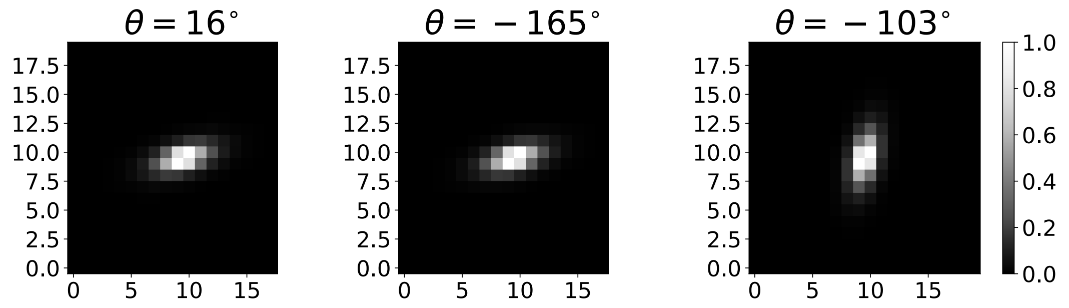


Figure 4.3: Examples of simulated galaxy images at different rotation angles, downsampled to 20×20 as in Izbicki et al. (2014). The left and center images show how angles that are π apart produce identical rotations, which is explored as conditional density estimation problem in setting (B) (see text).

Figure 4.4 provides the results for setting (A), in which the rotation angle $\theta \in [0, \pi]$. Figure 4.4, top, shows that all proposed models conditional estimation is significantly better than the marginal distribution in terms of CDE loss. If only the CDE loss is considered, one would choose the ResNet34 architecture using cosine basis functions as the best conditional density estimate. However, when looking at the distribution of the PIT statistics in the center panel of Figure 4.4, the PIT distribution for the three best models is not uniform. The PP-plot shows significant distributions from the expected uniform distribution, and running Kolmogorov-Smirnoff two-sample test with a uniformly distributed sample (Stephens, 1974) returns a p-value of $p < 1 \times 10^{-6}$. For the three best models the estimated conditional densities are overly broad, which is indicated by the PIT values being concentrated around 0.5. This is a due to the cosine basis expansion; cosine basis functions concentrate less sharply around the predicted angle than what the Haar basis does over the same interval. For this reason, the PIT statistics for the ResNet34 network using Haar basis are not broad, with the K-S test returning a non-significant p-value of $p = 0.23$. The bottom panel of Figure 4.4 shows representative examples of the difference between cosine and Haar basis functions in this example when using the deep architecture with the most capacity, ResNet34, with the true rotation angle indicated in red.

Figure 4.5 showcases the results for setting (B), in which the rotation angle $\theta \in [-\pi, \pi]$. As mentioned above, this makes the problem of correcting the rotation of symmetrical galaxies ill-posed (as two different angles can provide the same rotation), and so one would expect the conditional density to be bimodal. As in setting (A), all models achieve a CDE loss which is substantially smaller than the marginal distribution, with the models using cosine basis functions achieving the best results (Figure 4.5, top). However, as shown in the center panel of Figure 4.5, the PIT statistics for most models are not uniformly distributed, indicating issues with the conditional density fit. This can be explained by the tendency to overfit the CDE loss, which results in conditional density estimates which are either flat or unimodal. An example of this is provided in the bottom panel of Figure 4.5, where the ResNet18 model with a cosine basis function returns an uniformative conditional distribution in the first test case and a single peak in the second test case (true rotation angles indicated in red). The MLP model using the Haar basis, though not having the same capacity as other models, manages to correctly identifying the bimodal structure in the estimated conditional densities. In fact, the PIT statistics for the MLP model with Haar basis functions pass the uniformity test, with a p-value of $p = 0.11$ for the K-S two-sample test.

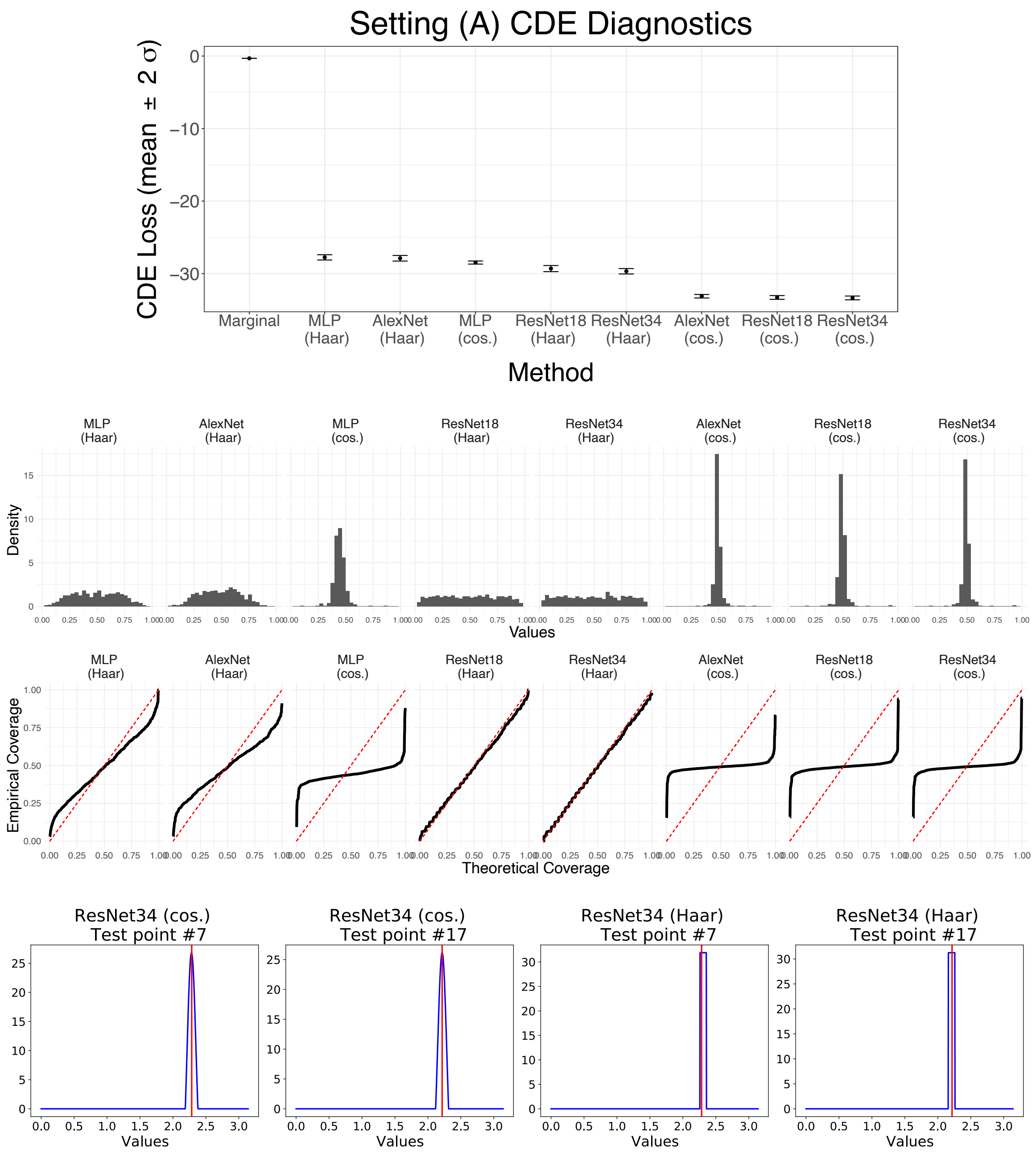


Figure 4.4: In setting (A), models using the cosine basis functions achieve a smaller CDE loss. However, the resulting conditional density estimates are too broad, with PIT statistics not uniformly distributed. Using Haar basis produces higher CDE loss values but better conditional density estimates. *Top*: Mean and 2 standard deviation values for the CDE loss of all conditional density models. *Center*: PIT statistic distribution and PP-plot against the expected uniform distribution. *Bottom*: Example conditional density estimates for the Resnet34 architecture with both cosine and Haar basis functions, with the true rotation angle indicated in red.

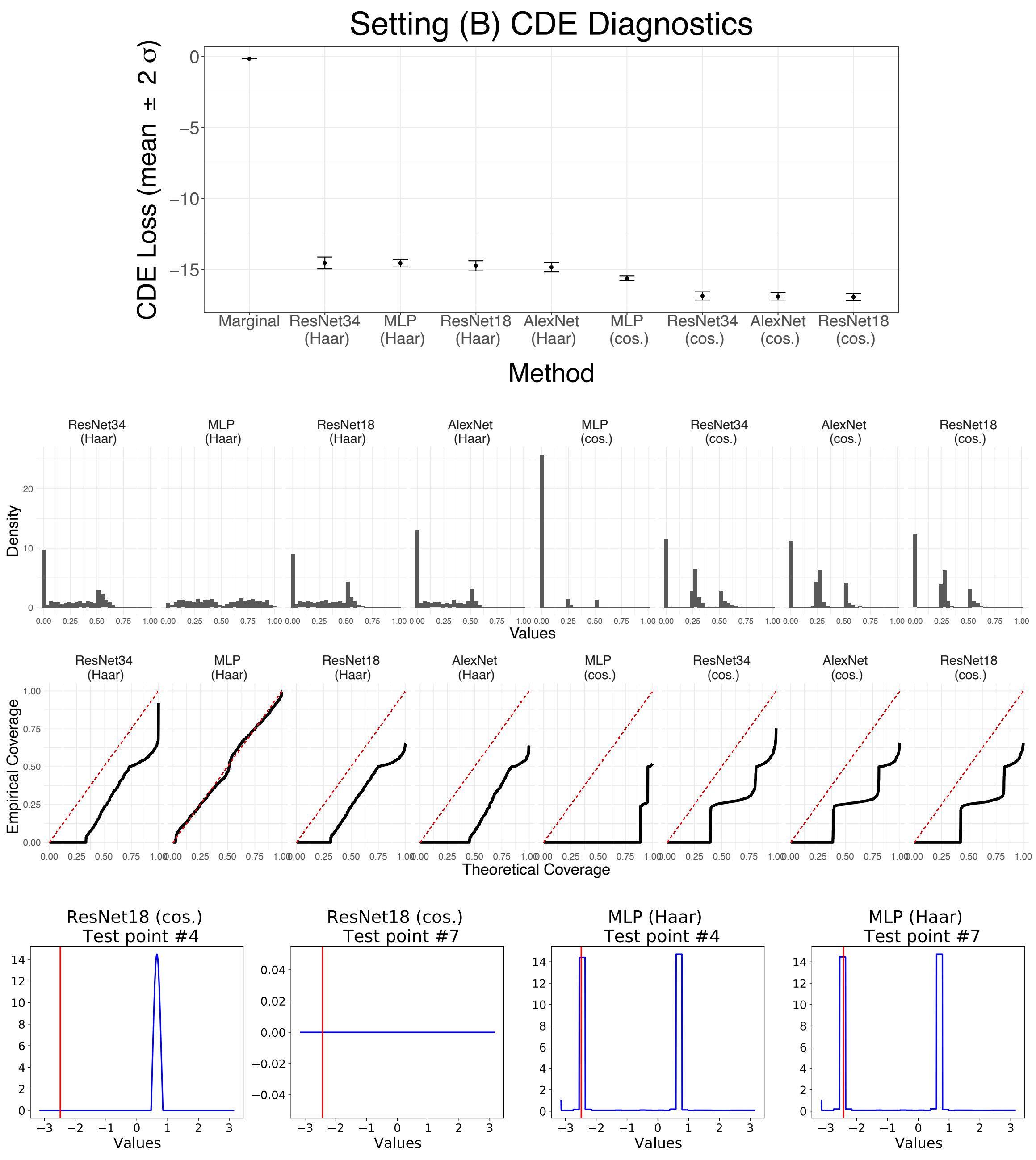


Figure 4.5: In setting (B), higher capacity models achieve a smaller CDE loss. However, the resulting conditional density estimations does not capture the bi-modal nature of the setting (see text). Most models in fact estimate a conditional density with a single peak, while the MLP with Haar basis correctly recovers the bimodal structure. *Top*: Mean and 2 standard deviation values for the CDE loss of all conditional density models. *Center*: PIT statistic distribution and PP-plot against the expected uniform distribution. *Bottom*: Example conditional density estimates for the Resnet18 model with cosine basis (best CDE loss) and MLP with Haar basis functions (only model for which the PIT statistic are distributed uniformly), with the true rotation angle indicated in red.

4.5 Application: Time-Series Prediction of Benzene Concentration

In this section we provide an example of estimating conditional densities when the input is a multivariate time series. We consider the problem of predicting the level of air pollution given multi-sensor chemical measurements of the air quality. More specifically we focus on the problem of predicting benzene concentration in the air, given the known link between cancer and long exposure to benzene in the air (Mage et al., 1996). We use the multi-sensors chemical measurements recorded in a polluted city in Italy by De Vito et al. (2008). The data include hourly measurements of the levels of carbon monoxide, ozone, nitrogen dioxide, generic nitrogen dioxides, benzene, as well as temperature and absolute and relative humidity, over the course of a 13 month period (March 2004 to April 2005). In this section we estimate the conditional distribution of benzene for a specific hour given the multivariate time series of hourly measurements in the 30 days before the specific date. In other words, the input to each model is a 7-dimensional time series with 720 time steps (once every hour for a month). We use the first 10 months of the data as a training set, and the remaining 3 as a test set. In terms of neural architectures we explore feed-forward MLPs with up to 3 hidden layers, as well as long-short term memory (LSTM) recurrent neural network (Hochreiter and Schmidhuber, 1997), which are specifically designed to handle sequential data. We consider a single LSTM layer, stacking two LSTM layers and a bi-directional LSTM, in which the sequence is fed in both directions to the neural architecture (Schuster and Paliwal, 1997). As in Section 4.4, we explore both the cosine basis and Haar basis functions for the conditional density basis expansions.

Figure 4.7 shows the results for the CDE fit of the different models and basis functions. In Figure 4.7, top, we see that all models achieve a CDE loss significantly lower than the marginal distribution, with the bi-directional LSTM achieving the lower loss values with both basis functions. However, when looking at the PIT and HPD diagnostics for bi-directional LSTM in the center and bottom panels of Figure 4.7, there tends to be an over-representation of low PIT and high HPD statistics, indicating an overly-narrow fit. Indeed, bi-directional fit do not pass the K-S two-sample test for uniformity for either PIT or HPD. On the other hand, a two-layers LSTM using cosine basis passes the uniformity test for both PIT and HPD, hence providing a well-calibrated conditional density estimate for benzene concentration. Figure 4.6 provides two representative examples of the difference between a bidirectional LSTM (left panels) and two-layers LSTM (right panel), where the true benzene value is indicated in red. The bidirectional LSTM tends to produce more sharp density estimates that in some cases assign zero probability to the actual response, while two-layer LSTM assign a non-zero probability to a larger region of the response space and are hence better calibrated.

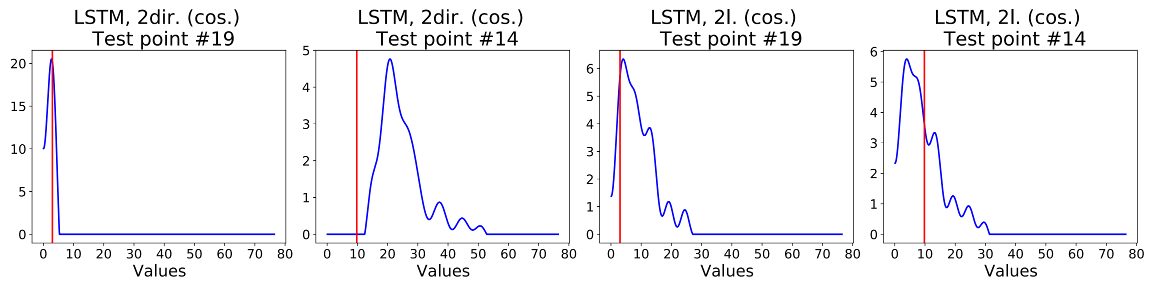


Figure 4.6: Examples of conditional density estimates for two test set hourly values (in red). The bidirectional LSTM provides sharper conditional density estimates that can sometime assign the true benzene concentration a probability zero, while the 2-layer LSTM is more broad and hence better calibrated.

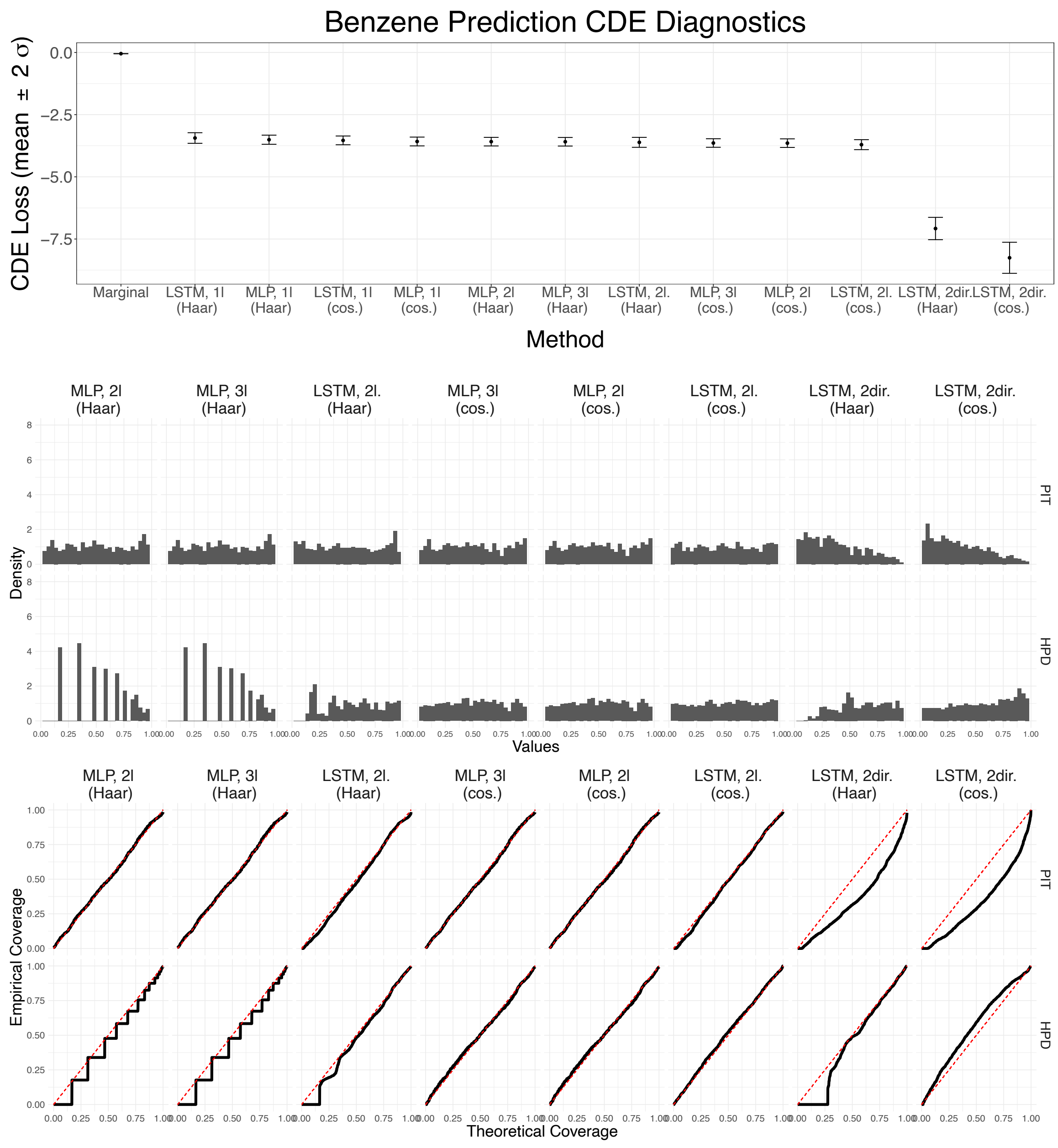


Figure 4.7: In benzene concentration prediction the bidirectional LSTM models achieve the lowest CDE loss, but the estimated conditional densities are overly narrow, as indicated by the PIT and HPD statistics. On the other hand, 2-layer LSTM do not achieve quite the same CDE loss, but provide better calibrated conditional densities. *Top*: Mean and 2 standard deviation values for the CDE loss of all conditional density models. *Center*: PIT and HPD statistic distribution. *Bottom*: PP-plot of PIT and HPD statistics against the expected uniform distribution.

Chapter 5

Conclusions and Future Work

This thesis develops tools with frequentist guarantees for high-dimensional simulator-based settings by leveraging machine learning algorithms. Firstly, we introduce a statistical framework for constructing valid confidence sets and hypothesis tests with finite sample guarantees. We provide two novel test statistics for likelihood-free frequentist inference called `ACORE` and `BFF`, which approximate the likelihood ratio and Bayes factor, respectively. We also develop practical and modular methods to estimate critical values and p-values, and to evaluate the empirical coverage of confidence sets across the parameter space. Secondly, we propose a statistically consistent approach for validating fitted approximate likelihood models which can pinpoint the locations in the parameter space where the fit is inadequate and provide insights as to how the high-resolution simulator and approximate likelihood model differ. Finally, we contribute a neural density estimation tool that transforms deep regression architectures in conditional density estimators with minimal computational overhead. Different extensions are possible, and this chapter reviews the main ones for the statistical framework presented in Chapter 2

5.1 Hyper-Parameter Tuning and Model Selection

In the applications showcased in Section 2.9, confidence sets are constructed using different probabilistic classification algorithms, which are then selected according to the lowest cross-entropy loss on a separate validation set. As mentioned in Section 2.9, the estimation of the likelihood seems to affect the power achieved by the test statistic significantly, especially in high-dimensions. Hence providing an accurate odds is critical to the properties of the downstream confidence sets. Overall, all algorithms are employed using their vanilla implementation (that is, the built-in standard setting of, mostly, `scikit-learn` (Pedregosa et al., 2011)), but one could use hyper-parameter optimization techniques to improve their fit, especially for what concerns high-capacity probabilistic classification methods such as deep neural network. Examples of potential hyper-parameter optimization algorithms are Bayesian methods (Snoek et al., 2012) or bandit-based approaches (Li et al., 2018), as well as more recent combinations of the two (Falkner et al., 2018), leveraging dedicated software such as `OPTUNA` (Akiba et al., 2019) or `SHERPA` (Hertel et al., 2020).

In all experiments, the cross-entropy loss and the L_2 odds loss proposed in Section 2.5 follow the same pattern in odds estimation with a large sample size (i.e., high values of B): the better the fit, the smaller the loss. However, the odds loss appears unstable, especially for small sample sizes (e.g., Figures 2.5 and 2.13), in which poorly estimated odds can achieve significantly large values. Nevertheless, estimating odds would ultimately provide good likelihood approximation, resulting in higher power for downstream inference. One could train a probabilistic classifier to directly minimize the L_2 odds, when feasible (such as gradient-boosted trees and deep neural networks). Instabilities in training could be handled by controlling training hyper-parameters such as learning rate and batch size (Bengio, 2012; Masters and Luschi, 2018), using batch-normalization to mitigate potentially disruptive weight initialization (Ioffe and Szegedy, 2015), as well as using gradient clipping to avoid exploding gradients (Pascanu et al., 2013).

5.2 Guided Simulations For Critical and P-Values Estimation

Critical and p-values estimation in the inference machinery described in Chapter 2 relies on a separate simulation sample $\mathcal{T}' = \{\theta_i, \tau(\mathcal{D}_{\theta_i}; \theta_i)\}_{i=1}^{B'}$ to determine the distribution of the test statistic in different regions of the parameter space Θ. In current experiments, the parameter space is explored uniformly in the $\mathcal{T}'$ sample, using a uniform proposal distribution r_Θ over the parameter space. However, this approach becomes quickly inefficient with larger parameter spaces and higher observed sample sizes n. Roughly speaking, the higher the dimension of the parameter space, the smaller in proportion the area around the data-generating parameter value becomes. A uniform sampling approach over-samples from regions of the parameter space where the distribution of the test statistic $\widehat{\tau}(\mathcal{D};\theta)$ is very different from the test statistic at the observed data $\widehat{\tau}(\mathrm{D};\theta)$. Sampling from such regions harms the estimation of critical values and p-values. For critical values, the α-quantile would be over-smoothed for relevant areas of the parameter space. For p-values, most of the indicator variables $Z = \mathbb{I}\left(\widehat{\tau}(\mathcal{D};\theta_0) < \widehat{\tau}(D;\theta_0)\right)$ would be zero, making the probabilistic classification task significantly harder. Both of these effects result in over-covering confidence sets, with p-values empirically being affected the most (see, e.g., Section 2.9.4).

To provide more efficient sampling across the parameter space, one could define a data-driven proposal distribution π over the parameter space Θ. The key idea is to use the value of $\widehat{\tau}(\mathrm{D};\theta)$, the estimated test statistic at each parameter value for the observed data, to guide the sampling of parameter values θ for the $\mathcal{T}'$ sample. With perfectly estimated odds, $\tau(\mathrm{D};\theta)$ would indeed be the largest at the true data-generating parameter θ. Algorithm 12 shows how the critical and p-values estimation algorithms would accommodate a guided simulation scheme using a Gaussian proposal distribution π. Firstly, we use a separate uniform sample of the parameter space to compute the estimated test statistic across the parameter space. Secondly, we normalize the test statistic values to define a discrete distribution over the parameter space, which we approximate with a Gaussian. Finally, we define the Gaussian approximation as the new proposal distribution π and use it to estimate critical and p-values as in Algorithm 4. Note that in the current setup the guided simulation is not an active learning strategy, as the Gaussian proposal distribution is estimated using a sample size K chosen a priori. One could envision an extension in which the Gaussian approximation and confidence intervals are constructed sequentially, which could tackled using the notion of confidence sequences (Darling and Robbins, 1967; Lai, 1976; Howard et al., 2021).

Algorithm 12 Estimate the p-values $p(D;\theta_0)$ or critical values C_{θ_0}, given observed data D, for a level-α test of $H_{0,\theta_0} : \theta = \theta_0$ vs. $H_{1,\theta_0} : \theta \neq \theta_0$ for all $\theta_0 \in \Theta$ simultaneously using Gaussian guided sampling.

Input: observed data D; stochastic forward simulator F_θ; sample size K for guided strategy; π (a fixed proposal distribution over the full parameter space Θ); test statistic $\widehat{\tau}$; regression estimator m; desired level $\alpha \in (0,1)$: confidence set estimation strategy `eval` (either `p-values` or `crit-values`)

Output: estimated p-value $\widehat{p}(D;\theta)$ or critical value $\widehat{C}_\theta$ for all $\theta = \theta_0 \in \Theta$

```
 1: // **Initial Sample from** $\Theta$
 2: Set $a_\theta \leftarrow \emptyset$ and $a_\tau \leftarrow \emptyset$
 3: **for** k in $\{1, \dots, K\}$ **do**
 4:     Draw parameter $\theta_k \sim \pi$
 5:     Compute test statistic $\widehat{\tau}_k \leftarrow \widehat{\tau}(D;\theta_k)$
 6:     Append to vectors $a_\theta \leftarrow a_\theta \cup \{\theta_k\}$ and $a_\tau \leftarrow a_\tau \cup \{\widehat{\tau}_k\}$
 7: **end for**
 8: // **Gaussian Approximation for** $\widehat{\tau}(D;\theta)$ **over** $\Theta$
 9: Normalize test statistics $a_\tau \leftarrow \frac{\exp(a_\tau)}{\sum_k \exp(a_{\tau,i})}$
10: Define discrete distribution $r_{a_\theta}$ s.t. $\mathbb{P}(a_{\theta,k}) = a_{\tau,k}$
11: Set $g_\theta \leftarrow \emptyset$
12: **for** j in $\{1, \dots, K\}$ **do**
13:     Draw parameter $\theta_j \sim r_{a_\theta}$
14:     $g_\theta \leftarrow g_\theta \cup \{\theta_j\}$
15: **end for**
16: Set $\mu = \frac{1}{B_1'} \sum_j g_{\theta,j}$ and $\Sigma = \frac{1}{B_1'} \sum_j (g_{\theta,j} - \mu)(g_{\theta,j} - \mu)^T$
17: // **P-Values or Critical Values Estimation**
18: **if** eval == crit-values **then**
19:     Estimated critical value $\widehat{C}_{\theta_0}$ via Algorithm 4 with $\pi = \mathcal{N}(\mu, \Sigma)$
20:     **return** $\widehat{C}_\theta$ for all $\theta = \theta_0 \in \Theta$
21: **else if** eval == p-values **then**
22:     Estimated p-value $\widehat{p}(D;\theta_0)$ via Algorithm 5 with $\pi = \mathcal{N}(\mu, \Sigma)$
23:     **return** $\widehat{p}(D;\theta)$ for all $\theta = \theta_0 \in \Theta$
24: **end if**
```

5.3 Efficient Integration and Optimization

As mentioned in Section 2.7, one of the primary sources of error to the test statistic power is the numerical error when computing the maximization or integration values at the denominator of the `ACORE` and `BFF` test statistics, respectively. As a preamble, integration and maximization have a strong connection, highlighted by the Laplace approximation (see

Barndorff-Nielsen and Cox 1989; Young et al. 2005 for a detailed summary). Suppose we have a function $h(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ and we wish to evaluate:

$$h_n = \int_{\mathcal{X}} e^{-nh(\mathbf{x})} d\mathbf{x},$$

for some $n > 0$. For large values of n, the largest contribution to the integral value will be around the maximum of $e^{-nh(\mathbf{x})}$, or at the minimum of $h(\mathbf{x})$ in $\mathcal{X}$. Approximating the integral with the maximum is an accurate approximation up to order $O(n^{-1})$. This fact is used in Bayesian inference to avoid explicit marginalization of variables in the posterior distribution (integration task) by computing the maximum a posteriori (maximization task); see (Bishop, 2006, Chapter 4) for a more detailed treatment.

In current experiments, maximization is computed by evaluating over an equispaced grid in the parameter space, which is accurate to order $O(n^{-1/d})$, while integration is performed via Monte Carlo integration, which is accurate to order $O_p(n^{-1})$.[*] However, there are a plethora of methods that could reduce the approximation error even further. For maximization tasks, the Laplace approximation is applicable since the likelihood ratio surface peaks exponentially at the true parameter value as a function of the observed sample size n. In other words, one could use a Gaussian approximation of the estimated odds and use the maximum likelihood estimator of the maximum in place of the integral, which converges as $O_p(n^{-1/2})$ (assuming $n \gg d$). Laplace approximations might not, however, be robust to estimation errors in learning the odds (e_1 in Section 2.7). Another approach can be borrowed from the hyper-parameter search literature for machine learning algorithms (see Feurer and Hutter (2019) for an overview). For instance, kernel-based Bayesian optimization can achieve an approximation of the order $O_p\left(\sqrt{\frac{dlog(n)}{n}}\right)$ (Kandasamy et al., 2015). For integration tasks, more recent methods have improved the convergence rates of Monte Carlo integration. For instance, adaptive methods achieved a convergence rate of $O_p\left(\frac{\log(n)^d}{n}\right)$ (Weinzierl, 2000), while Quasi-Monte Carlo methods provide a further improvement with a convergence rate of the order $O_p\left(\frac{\log(n)^d}{n^2}\right)$ (Dick et al., 2013). As a note, the Laplace approximation could also be exploited in the opposite direction, hence using integration to evaluate the maximum of a function.

[*]Note that this convergence rate is in probability, as Monte Carlo integration is stochastic, as well as independent of the dimension d (Weinzierl, 2000).

5.4 Approaches to High-dimensional Parameter Spaces

As shown in Section 2.8, one can approach high-dimensional parameter spaces by separating the parameters in parameters of interest Φ and nuisance parameters Ψ, that is, parameters which are not of interest in the inference process, so that $\Theta = \Phi \times \Psi$. However, nuisance parameter profiling requires evaluating the maximum of the likelihood for any values of $\phi \in \Phi$, which in frequentist likelihood-free inference is further complicated by the fact that either the likelihood or the likelihood gradient is not known. In other words, profiling in high-dimensional parameter spaces requires both (i) an efficient maximization algorithm and (ii) an accurate estimation of the likelihood or odds functions. In this section, we provide a summary of observations and potential approaches to handle high-dimensional parameter spaces.

The high-energy physics literature provides a series of relevant approaches to high-dimensional parameter spaces. The goal is to de-correlate nuisance parameters and observed data, reducing the effect of nuisance parameters on observable data. In fact, in situations where the nuisance parameter has a negligible impact on the observable data, one could disregard the nuisance parameters or opt for more computationally friendly profiling approaches, such as setting the nuisance parameters at the maximizer of the likelihood. One approach to reduce nuisance parameter effects is planing (Chang et al., 2018). Planing is a data pre-processing techniques in which the training data are re-weighted so that they attain the same probability distribution function across nuisance parameters. However, this technique requires an exact knowledge or accurate approximation of the likelihood, and might not scale efficiently in higher-dimension. Another approach is to include an explicit penalty term in the loss function used to estimate the likelihood to enforce insensitivity to nuisance parameters (Stevens and Williams, 2013; Kasieczka and Shih, 2020). Such penalty terms would also be applicable for learning the odds in Section 2.2. Another stream of work employs adversarial schemes in order to train neural networks to develop likelihood estimates that are de-correlated from nuisance parameters (Louppe et al., 2017) and could be used as test statistics for frequentist likelihood-free inference.

Another potential approach for frequentist likelihood-free inference in problems with a high-dimensional parameter space is to construct confidence sets to transform the parameters of interest $h(\theta)$ instead of the parameter θ itself. An example would be the ill-posed inverse problem tackled by Patil et al. (2020), in which the quantity of interest is the vertically integrated CO_2 concentration can be expressed as a linear transformation $\xi = w^T\theta$, with w being a vector of known weights and θ being the CO_2 concentrations at different altitudes. Constructing confidence sets would then require to sample different values of $\theta_1, ..., \theta_m \sim r_\Theta$, generate the simulated data $\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,n} \sim F_{\theta_i}$ and then associate each simulated dataset with the integrated average $\xi_i = h(\theta_i)$ as a new parameter value. However, a caveat in this setting is that the new implicit likelihood $\mathcal{L}(\mathbf{x}; \xi)$ is not guaranteed to be identifiable. In other words, if two parameter values θ_1, θ_2, such that $\theta_1 \neq \theta_2$ and $F_{\theta_1} \neq F_{\theta_2}$, attain the same integrated average $\xi = h(\theta_1) = h(\theta_2)$, the likelihood $\mathcal{L}(\mathbf{x}; \xi)$ would not be a well defined function, leading to degeneracies and artifacts in the downstream inferential results.

Finally, different approaches can be used to increase the efficiency of maximization and integration under the presence of nuisance parameters. For maximization tasks, one could use gradient-free optimization techniques (see Conn et al. 2009 and references therein), approximate the gradient by evaluating the simulators at closeby points, which can be used within gradient-based optimization methods (see Boyd and Vandenberghe 2004 for a review) or machine-learning specific optimization techniques mentioned in Section 5.3. For integration tasks, one could include prior knowledge to separate the joint distribution of nuisance parameters $\pi(\psi)$ into marginals and conditionals distributions, hence reducing the high-dimensional integral into a series of nested integral calculations.

5.5 Application to Epidemiological Models

Compartmental models in epidemiology are used to study the evolution of a disease across a population. Compartmental models can be cast either as a deterministic or stochastic system, using ordinary or stochastic differential equations, respectively. The most famous compartment model is the SIR model (SIR; McKendrick 1925), which divides the population into three different groups: susceptible (R), infectious (I) and removed (R). The sum of the three groups is the total population $N = S+I+R$, where N is usually a fixed, known number. There are many extensions to the SIR model and many successful applications: see Keeling and Rohani (2008) and Hethcote (2000) for a more systematic reviews and Chang et al. (2021) for a recent application on modeling COVID-19 spread dynamics. In this section, we will focus on stochastic compartmental models as they fall within the likelihood-free inference framework, as also shown in Radev et al. (2020).

Borrowing the same notation as in Greenwood and Gordillo (2009), let S_t, I_t and R_t be the population in the three different categories at time $t \in T$; as a reminder, here $S_t + I_t + R_t = N$ for all $t \in T$, so one can model S_t and I_t and fully characterize the population considered. In the time interval $[t + \Delta t]$:

- the probability that an individual is infected, so $S_t = S_{t-1} - 1$ and $I_t = I_{t-1} + 1$, is equal to $\beta \frac{SI}{N} \Delta t$, where β is the infection rate. One can interpret this probability as each infective individual potentially affecting the susceptible population with a rate β. (The denominator here can be changed to reflect inhomogeneous infections as shown by Stroud et al. 2006);

- the probability that an individual recovers, so $R_t = R_{t-1} + 1$ and $I_t = I_{t-1} - 1$, is $\gamma I \Lambda t$, where γ is a recovery rate.

Extending the above to a population level, each increment can be expressed as a sum of the expected increment and a stochastic term, which can be written as:

$$
\begin{aligned}
\Delta S &= \left(-\beta \frac{S_t I_t}{N} \right) \Delta t + \Delta Z_1 \\
\Delta I &= \left(\beta \frac{S_t I_t}{N} - \gamma I_t \right) \Delta t - \Delta Z_1 + \Delta Z_2
\end{aligned}
$$

where ΛZ_1 and ΛZ_2 are Poisson distributions such that $\mathbb{E}[Z_1] = \mathbb{E}[Z_2] = 0$, while $\mathrm{Var}[Z_1] = \beta \frac{S_t I_t}{N} \Delta t$ and $\mathrm{Var}[Z_2] = \gamma I_t \Delta t$. From a likelihood-free inference perspective, the SIR model has two parameters $\boldsymbol{\theta} = (\beta, \gamma)$ and the observable data are a two-dimensional time series $\{\mathbf{X}_t\} = (S_t, I_t)\}_{t=1}^{T}$. Note that, although there are three subcategories, two terms are enough to characterize the entire time series as $S_t + I_t + R_t = N$ for all $t \in T$. Within our framework, one could estimate the odds by either (i) flattening the time series and use the $2T$ values as tabular features in a probabilistic classifier (in, e.g., random forest or gradient boosted trees), (ii) extracting meaningful summary statistics as input to any probabilistic classifier or (iii) using the 2-D time series directly as inputs to recurrent neural networks such as LSTMs (Hochreiter and Schmidhuber, 1997). In cases when the time series are not regular, such as sampled at different intervals or with different values of T, one can rely on recent developments on recurrent neural ordinary equations (Rubanova et al., 2019) to learn the odds.

Bibliography

Aad, G., Abajyan, T., Abbott, B., Abdallah, J., Abdel Khalek, S., Abdelalim, A., Abdinov, O., Aben, R., Abi, B., Abolins, M., and et al. (2012). Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29. 2, 36, 71

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. 163

Abbott, T., Abdalla, F. B., Allam, S., Amara, A., Annis, et al. (2016). Cosmology from cosmic shear with Dark Energy Survey Science Verification data. *Physical Review D*, 94(2):022001. 72

Abbott, T. M. C., Alarcon, A., Allam, S., Andersen, P., Andrade-Oliveira, F., Annis, J., Asorey, J., Avila, S., Bacon, D., Banik, N., Bassett, B. A., Baxter, E., Bechtol, K., Becker, M. R., and et.al (2019). Cosmological constraints from multiple probes in the dark energy survey. *Phys. Rev. Lett.*, 122:171301. 5, 94

Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A. J., Barreiro, R. B., Bartolo, N., and et al. (2020). Planck 2018 results. VI. Cosmological parameters. *Astronomy & Astrophysics*, 641:A6. 94

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery. 118

Alsing, J., Charnock, T., Feeney, S., and Wandelt, B. (2019). Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 488(3):4440–4458. 3, 106

Ayano, T. (2012). Rates of convergence for the k-nearest neighbor estimators with smoother regression functions. *Journal of Statistical Planning and Inference*, 142(9):2530–2536. 32

Baker, A., Hammerling, D., Levy, M., Xu, H., Dennis, J., Eaton, B., Edwards, J., Hannay, C., Mickelson, S., Neale, R., et al. (2015a). A new ensemble-based consistency test for the Community Earth System Model. *Geoscientific Model Development Discussions*, 8(9). 72

Baker, A. H., Hammerling, D. M., Levy, M. N., Xu, H., Dennis, J. M., Eaton, B. E., Edwards, J., Hannay, C., Mickelson, S. A., Neale, R. B., Nychka, D., Shollenberger, J., Tribbia, J., Vertenstein, M., and Williamson, D. (2015b). A new ensemble-based consistency test for the community earth system model (pycect v1.0). *Geoscientific Model Development*, 8(9):2829–2840. 6

Baringhaus, L. and Franz, C. (2004). On a new multivariate two-sample test. *J. Multivar. Anal.*, 88(1):190–206. 83

Barlow, R. and Beeston, C. (1993). Fitting using finite monte carlo samples. *Computer Physics Communications*, 77(2):219 – 228. 3

Barndorff-Nielsen, O. E. and Cox, D. R. (1989). *Asymptotic Techniques for Use in Statistics.* Springer. 121

Beaumont, M. and Rannala, B. (2004). The Bayesian revolution in genetics. *Nature reviews. Genetics*, 5:251–61. 3

Beaumont, M. A. (2019). Approximate bayesian computation. *Annual Review of Statistics and Its Application*, 6(1):379–403. [3]

Bengio, Y. (2012). *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg. [118]

Bickel, P. J. and Li, B. (2007). Local polynomial regression on unknown manifolds. *Complex datasets and inverse problems*, pages 177–186. [32]

Bierens, H. J. (1983). Uniform consistency of kernel estimators of a regression function under generalized conditions. *Journal of the American Statistical Association*, 78(383):699–707. [29]

Birkes, D. (1990). Generalized likelihood ratio tests and uniformly most powerful tests. *The American Statistician*, 44(2):163–166. [47]

Bishop, C. M. (1994). *Mixture density networks.* Technical Report. [106]

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag, Berlin, Heidelberg. [121]

Blaker, H. (2000). Confidence curves and improved exact confidence intervals for discrete distributions. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 28(4):783–798. [24]

Bordoloi, R., Lilly, S. J., and Amara, A. (2010). Photo-z performance for precision cosmology. *Monthly Notices of the Royal Astronomical Society*, 406(2):881–895. [24]

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press. [123]

Brehmer, J., Cranmer, K., Louppe, G., and Pavez, J. (2018). Constraining effective field theories with machine learning. *Physical Review Letters*, 121(11). [3], [7], [10]

Brehmer, J., Kling, F., Espejo, I., and Cranmer, K. (2020a). MadMiner: Machine learning-based inference for particle physics. *Comput. Softw. Big Sci.*, 4(1):3. [64]

Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. (2020b). Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy of Sciences*, 117(10):5242–5249. 4, 5, 7, 10, 93

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. 97

Carrasco Kind, M. and Brunner, R. J. (2013). TPZ: photometric redshift PDFs and ancillary information by using prediction trees and random forests. *Monthly Notices of the Royal Astronomical Society*, 432(2):1483–1501. 97

Carrasco Kind, M. and Brunner, R. J. (2014). Sparse representation of photometric redshift probability density functions: preparing for petascale astronomy. *Mon Not R Astron Soc*, 441(4):3550–3561. 101

Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA. 24

Chang, S., Cohen, T., and Ostdiek, B. (2018). What is the machine learning? *Phys. Rev. D*, 97:056009. 122

Chang, S., Pierson, E., Koh, P., Gerardin, J., Redbird, B., Grusky, D., and Leskovec, J. (2021). Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589:1–6. 124

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. 101

Chen, Y. and Gutmann, M. U. (2019). Adaptive gaussian copula ABC. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1584–1592. PMLR. 4, 94

Chuang, C.-S. and Lai, T. L. (2000). Hybrid resampling methods for confidence intervals. *Statistica Sinica*, 10(1):1–33. 9, 35

Chwialkowski, K., Ramdas, A., Sejdinovic, D., and Gretton, A. (2015). Fast two-sample testing with analytic representations of probability measures. In *Proceedings of the 28th*

International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, pages 1981–1989, Cambridge, MA, USA. MIT Press. 77

Claeskens, G., Van Keilegom, I., et al. (2003). Bootstrap confidence bands for regression curves and their derivatives. *The Annals of Statistics*, 31(6):1852–1884. 24

Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, USA. 123

Connors, A., Esch, D. N., Freeman, P., Kang, H., Karovska, M., Kashyap, V., Siemiginowska, A., Zezas, A., and van Dyk, D. (2006). Deconvolution in high-energy astrophysics: science, instrumentation, and methods. *Bayesian Analysis*, 1(2):189 – 235. 6

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692. 7, 24, 81

Cousins, R. D. (2018). Lectures on Statistics in Theory: Prelude to Statistics in Practice. *arXiv e-prints*, page arXiv:1807.05996. 7, 9, 11

Cousins, R. D. and Highland, V. L. (1992). Incorporating systematic uncertainties into an upper limit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 320(1-2):331–335. 36

Cranmer, K. (2015). Practical Statistics for the LHC. *CERN Technical Report*, pages 267–308. 41 p. 7, 9

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062. 2, 9, 61, 93

Cranmer, K., Pavez, J., and Louppe, G. (2015). Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint 1506.02169*. 4, 43, 63

Cunha, C. E., Lima, M., Oyaizu, H., Frieman, J., and Lin, H. (2009). Estimating the redshift distribution of photometric galaxy samples–II. applications and tests of a new method. *Monthly Notices of the Royal Astronomical Society*, 396(4):2379–2398. 97

Dalmasso, N., Izbicki, R., and Lee, A. (2020a). Confidence Sets and Hypothesis Testing in a Likelihood-Free Inference Setting. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2323–2334. PMLR. 8

Dalmasso, N., Lee, A., Izbicki, R., Pospisil, T., Kim, I., and Lin, C.-A. (2020b). Validation of Approximate Likelihood and Emulator Models for Computationally Intensive Simulations. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3349–3361, Online. PMLR. 8

Dalmasso, N., Pospisil, T., Lee, A. B., Izbicki, R., Freeman, P. E., and Malz, A. I. (2020). Conditional density estimation tools in Python and R with applications to photometric redshifts and likelihood-free cosmological inference. *Astronomy and Computing*, 30:100362. 8

Dalmasso, N., Zhao, D., Izbicki, R., and Lee, A. B. (2021). Likelihood-free Frequentist Inference. *In preparation.* 8

Darling, D. and Robbins, H. (1967). Confidence sequences for mean, variance, and median. *Proceedings of the National Academy of Sciences of the United States of America*, 58 1:66–8. 119

De Castro, P. and Dorigo, T. (2019). Inferno: Inference-aware neural optimisation. *Computer Physics Communications*, 244:170–179. xxi, 51, 52, 53

De Vito, S., Massera, E., Piga, M., Martinotto, L., and Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757. 114

Devroye, L., Győrfi, L., and Lugosi, G. (2013). *A Probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media. 32

Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288. 121

Dietrich, J. P. and Hartlap, J. (2010). Cosmology with the shear-peak statistics. *Monthly Notices of the Royal Astronomical Society*, 402(2):1049–1058. 86

Diggle, P. J. and Gratton, R. J. (1984). Monte Carlo Methods of Inference for Implicit Statistical Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):193–227. 2, 71

D'Isanto, A. and Polsterer, K. (2018). Photometric redshift estimation via deep learning-generalized and pre-classification-less, image based, fully probabilistic redshifts. *Astronomy & Astrophysics*, 609:A111. 106

Donoho, D. L. (1994). Asymptotic minimax risk for sup-norm loss: solution via optimal recovery. *Probability Theory and Related Fields*, 99(2):145–170. 29

Döring, M., Györfi, L., and Walk, H. (2017). Rate of convergence of k-nearest-neighbor classification rule. *J. Mach. Learn. Res.*, 18(1):8485–8500. 64, 65

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, New York, 2 edition. 161

Durkan, C., Murray, I., and Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2771–2781. PMLR. 4

D'Isanto, A. (2016). Uncertain photometric redshifts with deep learning methods. *Proceedings of the International Astronomical Union*, 12(S325):209–212. 104

Edlund, S., Davis, M., and Kaufman, J. (2010). The spatiotemporal epidemiological modeler. 1

Engl, H. W., Hanke, M., and Neubauer, A. (1996). *Regularization of inverse problems*, volume 375. Springer Science & Business Media. 6

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158. 98

Eubank, R. L. and Speckman, P. L. (1993). Confidence bands in nonparametric regression. *Journal of the American Statistical Association*, 88(424):1287–1301. 24

Evans, S. N. and Stark, P. B. (2002). Inverse problems as statistics. *Inverse Problems*, 18(4):R55–R97. 5

Falkner, S., Klein, A., and Hutter, F. (2018). BOHB: Robust and efficient hyperparameter optimization at scale. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1437–1446, Stockholmsmässan, Stockholm Sweden. PMLR. 118

Fasiolo, M., Wood, S. N., Hartig, F., and Bravington, M. V. (2018). An extended empirical saddlepoint approximation for intractable likelihoods. *Electron. J. Statist.*, 12(1):1544–1578. 4, 71

Feldman, G. (2000). Multiple measurements and parameters in the unified approach. Technical report, Technical Report, Talk at the FermiLab Workshop on Confidence Limits. 35

Feldman, G. J. and Cousins, R. D. (1998). Unified approach to the classical statistical analysis of small signals. *Physical Review D*, 57(7):3873–3889. 7, 9, 10

Feurer, M. and Hutter, F. (2019). *Hyperparameter Optimization*, pages 3–33. Springer International Publishing, Cham. 121

Fisher, R. (1925). *Statistical methods for research workers.* Oliver and Boyd: Edinburgh, 11th ed. rev. edition. 9

Frate, M., Cranmer, K., Kalia, S., Vand enberg-Rodes, A., and Whiteson, D. (2017). Modeling Smooth Backgrounds and Generic Localized Signals with Gaussian Processes. *arXiv e-prints*, page arXiv:1709.05681. 7, 10, 43, 61

Freeman, P. E., Izbicki, R., and Lee, A. B. (2017). A unified framework for constructing, tuning and assessing photometric redshift density estimates in a selection bias setting. *Monthly Notices of the Royal Astronomical Society*, 468(4):4556–4565. 95

Friedman, J. (2004). On multivariate goodness-of-fit and two-sample testing. Technical report, Stanford Linear Accelerator Center, Menlo Park, CA (US). 89

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232. 87

Gao, W. and Zhou, Z.-H. (2020). Towards convergence rate analysis of random forests for classification. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9300–9311. Curran Associates, Inc. 32

Genest, C. and Rivest, L.-P. (2001). On the multivariate probability integral transformation. *Statistics & Probability Letters*, 53(4):391–399. 104

Ghosh, J. K. (1961). On the relation among shortest confidence intervals of different types. *Calcutta Statistical Association Bulletin*, 10(4):147–152. 41

Girard, S., Guillou, A., and Stupfler, G. (2014). Uniform strong consistency of a frontier estimator using kernel regression on high order moments. *ESAIM: Probability and Statistics*, 18:642–666. 29

Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., Greenberg, D. S., and Macke, J. H. (2019). Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*. 3

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press. 106

Gourieroux, C., Monfort, A., and Renault, E. (1993). Indirect inference. *Journal of Applied Econometrics*, 8:S85–S118. 3

Gramacy, R. B. (2020). *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida. `http://bobby.gramacy.com/surrogates/`. 6

Greenberg, D., Nonnenmacher, M., and Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2404–2414, Long Beach, California, USA. PMLR. 4, 94

Greenwood, P. E. and Gordillo, L. F. (2009). *Stochastic Epidemic Modeling*, pages 31–52. Springer Netherlands, Dordrecht. 124

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773. 77, 83

Gupta, A., Matilla, J. M. Z., Hsu, D., and Haiman, Z. (2018). Non-Gaussian information from weak lensing data via deep learning. *Physical Review D*, 97(10):103515. 72

Gutmann, M. U., Cor, J., and er (2016). Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47. 4

Győrfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006). *A Distribution-Free Theory of Nonparametric Regression*. Springer Science & Business Media. 32

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, pages 331–371. 24, 109

Hardle, W., Luckhaus, S., et al. (1984). Uniform consistency of a class of regression function estimators. *The Annals of Statistics*, 12(2):612–623. 29

Harrison, D., Sutton, D., Carvalho, P., and Hobson, M. (2015). Validation of Bayesian posterior distributions using a multidimensional Kolmogorov–Smirnov test. *Monthly Notices of the Royal Astronomical Society*, 451(3):2610–2624. 104

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. 59, 109

Heinrich, J. (2003). Coverage of error bars for poisson data. *Technical Report*. 24

Hermans, J., Begy, V., and Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. *arXiv preprint 1903.04057*. 4

Hertel, L., Collado, J., Sadowski, P., Ott, J., and Baldi, P. (2020). Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, 12:100591. 118

Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653. 124

Hildebrandt, H., Viola, M., Heymans, C., Joudaki, S., Kuijken, K., et al. (2017). KiDS-450: cosmological parameter constraints from tomographic weak gravitational lensing. *Monthly Notices of the Royal Astronomical Society*, 465:1454–1498. 72

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780. 114, 125

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. (2021). Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055 – 1080. 119

Hu, J. and Bai, Z. (2016). A review of 20 years of naive tests of significance for high-dimensional mean vectors and covariance matrices. *Science China Mathematics*, 59(12):2281–2300. 77

Hurrell, J. W., Holland, M. M., Gent, P. R., and Ghan, S. (2013). The community Earth system model: a framework for collaborative research. *Bulletin of the American Meteorological Society*, 94(9):1339–1360. 1, 5, 71

Hyndman, R. J. (1996). Computing and graphing highest density regions. *The American Statistician*, 50(2):120–126. 104

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR. 118

Izbicki, R., Lee, A., and Schafer, C. (2014). High-dimensional density ratio estimation with extensions to approximate likelihood computation. In *Artificial Intelligence and Statistics*, pages 420–429. xxiv, 4, 59, 95, 109, 110

Izbicki, R. and Lee, A. B. (2016). Nonparametric conditional density estimation in a high-dimensional regression setting. *Journal of Computational and Graphical Statistics*, 25(4):1297–1316. 101, 102, 108

Izbicki, R. and Lee, A. B. (2017). Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics*, 11(2):2800–2831. 99, 101

Izbicki, R., Lee, A. B., and Freeman, P. E. (2017). Photo-z estimation: An example of nonparametric conditional density estimation under selection bias. *The Annals of Applied Statistics*, 11(2):698–724. 95, 104

Izbicki, R., Lee, A. B., and Pospisil, T. (2019). ABC–CDE: Toward approximate bayesian computation with complex high-dimensional data and limited simulations. *Journal of Computational and Graphical Statistics*, pages 1–20. 4, 94, 95, 103

Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(2):203–222. 12, 14

Jeffreys, H. (1961). *Theory of probability*. Clarendon Press Oxford, 3rd ed. edition. 12, 14

Jitkrittum, W., Szabó, Z., Chwialkowski, K. P., and Gretton, A. (2016). Interpretable distribution features with maximum testing power. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 181–189. Curran Associates, Inc. 77

Joudaki, S., Blake, C., Johnson, A., Amon, A., Asgari, M., Choi, A., Erben, T., Glazebrook, K., Harnois-Déraps, J., Heymans, C., Hildebrandt, H., Hoekstra, H., Klaes, D., Kuijken, K., Lidman, C., Mead, A., Miller, L., Parkinson, D., Poole, G. B., Schneider, P., Viola, M., and Wolf, C. (2017). KiDS-450 + 2dFLenS: Cosmological parameter constraints from weak gravitational lensing tomography and overlapping redshift-space galaxy clustering. *Monthly Notices of the Royal Astronomical Society*, 474(4):4894–4924. 94

Juric, M., Axelrod, T., Becker, A. C., Becla, J., Bellm, E., Bosch, J. F., Ciardi, D., Connolly, A. J., Dubois-Felsmann, G. P., Economou, F., Freemon, M., Gelman, M., Graham, M., Ivezić, Ž., Jenness, T., Kantor, J., Krughoff, K., Lim, K.-T., Lupton, R. H., Mueller, F., Nidever, D., Patterson, M., Petravick, D., Shaw, D., Slater, C., Strauss, M., Swinbank, J., Tyson, J. A., Wood-Vasey, M., and Wu, X. (2017). LSST data Products Definition Document. 101

Järvenpää, M., Gutmann, M. U., Pleska, A., Vehtari, A., and Marttinen, P. (2019). Efficient acquisition rules for model-based approximate bayesian computation. *Bayesian Anal.*, 14(2):595–622. 4

Järvenpää, M., Gutmann, M. U., Vehtari, A., and Marttinen, P. (2021). Parallel gaussian process surrogate bayesian inference with noisy likelihood evaluations. *Bayesian Anal.*, 16(1):147–178. 4

Kacprzak, T., Kirk, D., Friedrich, O., Amara, A., Refregier, A., et al. (2016). Cosmology constraints from shear peak statistics in Dark Energy Survey Science verification data. *Monthly Notices of the Royal Astronomical Society*, 463(4):3653–3673. 5, 72, 86

Kaipio, J. and Somersalo, E. (2007). Statistical inverse problems: Discretization, model reduction and inverse crimes. *J. Comput. Appl. Math.*, 198(2):493–504. 6

Kanamori, T., Suzuki, T., and Sugiyama, M. (2012). f-divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory*, 58(2):708–720. 77

Kandasamy, K., Schneider, J., and Poczos, B. (2015). High dimensional bayesian optimisation and bandits via additive models. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 295–304, Lille, France. PMLR. 121

Karabatsos, G. and Leisen, F. (2018). An approximate likelihood perspective on ABC methods. *Statistics Surveys*, 12(none):66 – 104. 3

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20. 162

Kasieczka, G. and Shih, D. (2020). Robust jet classifiers through distance correlation. *Physical Review Letters*, 125(12). 122

Kay, J., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., Arblaster, J., Bates, S., Danabasoglu, G., Edwards, J., Holland, M., Kushner, P., Lamarque, J.-F., Lawrence, D., Lindsay, K., Middleton, A., Munoz, E., Neale, R., Oleson, K., and Vertenstein, M. (2015). The Community Earth System Model (CESM) large ensemble project: a community resource for studying climate change in the presence of internal climate Variability. *Bull. Amer. Meteor. Soc.* 72

Keeling, M. J. and Rohani, P. (2008). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press. 123

Kim, I., Lee, A. B., Freeman, P., and Newman, J. (2016). Comparing distributions of galaxy morphologies. Technical report, Carnegie Mellon University, Department of Statistics & Data Science. 73

Kim, I., Lee, A. B., and Lei, J. (2019). Global and local two-sample tests via regression. *Electronic Journal of Statistics*, 13(2):5253 – 5305. 73, 78, 79, 85, 90, 91

Kim, I., Ramdas, A., Singh, A., and Wasserman, L. (2021). Classification accuracy as a proxy for two-sample testing. *The Annals of Statistics*, 49(1):411 – 434. 7, 77, 81

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4743–4751, USA. Curran Associates Inc. 5

Koenker, R., Chernozhukov, V., He, X., and Peng, L. (2017). *Handbook of quantile regression*. CRC press. 21

Kpotufe, S. (2011). k-NN Regression Adapts to Local Intrinsic Dimension. *Advances in Neural Information Processing Systems*, pages 729–737. 31

Kpotufe, S. and Garg, V. (2013). Adaptivity to local smoothness and dimension in kernel regression. *Advances in Neural Information Processing Systems*, 26:3075–3083. 32

Krause, E., Eifler, T., Zuntz, J., Friedrich, O., Troxel, M., Dodelson, S., Blazek, J., Secco, L., MacCrann, N., Baxter, E., et al. (2017). Dark energy survey year 1 results: Multi-probe methodology and simulated likelihood analyses. *arXiv preprint 1706.09359*. 94

Krivobokova, T., Kneib, T., and Claeskens, G. (2010). Simultaneous confidence bands for penalized spline estimators. *Journal of the American Statistical Association*, 105(490):852–863. 24

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc. 59, 108, 109

Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, 28(5):1–26. 162

Lai, T. L. (1976). On confidence sequences. *The Annals of Statistics*, 4(2):265–280. 119

Lall, U. and Sharma, A. (1996). A nearest neighbor bootstrap for resampling hydrologic time series. *Water Resources Research*, 32:679–693. 162

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436. 106

Lee, W.-C., Huang, H., Feng, G., Sanes, J., Brown, E., So, P., and Nedivi, E. (2006). Dynamic remodeling of dendritic arbors in gabaergic interneurons of adult visual cortex. *PLoS biology*, 4:e29. 1

Lewis, A. and Bridle, S. (2002). Cosmological parameters from cmb and other data: A monte carlo approach. *Physical Review D*, 66(10). 3

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52. 118

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22. 162

Liero, H. (1989). Strong uniform consistency of nonparametric regression function estimates. *Probability theory and related fields*, 82(4):587–614. 29

Lin, C.-A. and Kilbinger, M. (2015). A new model to predict weak-lensing peak counts-I. comparison with N-body simulations. *Astronomy & Astrophysics*, 576:A24. 86

Lopez-Paz, D. and Oquab, M. (2017). Revisiting classifier two-sample tests. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. xxiii, 7, 77, 81, 84, 85

Lorenz, R., Pitman, A., and Sisson, S. A. (2016). Does Amazonian deforestation cause global effects; can we be sure? *Journal of Geophysical Research: Atmospheres*, 121(10):5567–5584. 74

Louppe, G., Hermans, J., and Cranmer, K. (2019). Adversarial variational optimization of non-differentiable simulators. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1438–1447. PMLR. 4

Louppe, G., Kagan, M., and Cranmer, K. (2017). Learning to pivot with adversarial networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 122

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., and Macke, J. H. (2019). Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. 4, 94

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 1289–1299. Curran Associates, Inc. 4, 94

M. Tran, M.-N., Nott, D. J., and Kohn, R. (2017). Variational Bayes With Intractable Likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882. 4

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA. 25

Mage, D., Ozolins, G., Peterson, P., Webster, A., Orthofer, R., Vandeweerd, V., and Gwynne, M. (1996). Urban air pollution in megacities of the world. *Atmospheric environment*, 30(5):681–686. 114

Malz, A. I., Marshall, P. J., DeRose, J., Graham, M. L., Schmidt, S. J., Wechsler, R., and Collaboration), L. D. E. S. (2018). Approximating Photo- z PDFs for Large Surveys. *AJ*, 156(1):35. 102

Marin, J.-M., Raynal, L., Pudlo, P., Ribatet, M., and Robert, C. (2016). ABC random forests for bayesian parameter inference. *Bioinformatics (Oxford, England)*, 35. 4, 94

Masters, D. and Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *arXiv e-prints*, page arXiv:1804.07612. 118

Matthies, H. G. (2007). Quantifying uncertainty: Modern computational representation of probability and applications. In Ibrahimbegovic, A. and Kozar, I., editors, *Extreme Man-Made and Natural Hazards in Dynamics of Structures*, pages 105–135, Dordrecht. Springer Netherlands. 5

McKendrick, A. G. (1925). Applications of mathematics to medical problems. *Proceedings of the Edinburgh Mathematical Society*, 44:98–130. 123

Meeds, E. and Welling, M. (2014). GPS-ABC: Gaussian Process Surrogate Approximate Bayesian Computation. *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, page 593–602. 4, 71

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, 7(35):983–999. 21, 27, 97

Mohamed, S. and Lakshminarayanan, B. (2016). Learning in Implicit Generative Models. *arXiv e-prints*, page arXiv:1610.03483. 4, 80

Murphy, S. A. and Van Der Vaart, A. (2000). On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465. 36

Mustafa, M., Bard, D., Bhimji, W., Lukić, Z., Al-Rfou, R., and Kratochvil, J. (2019). Cosmogan: creating high-fidelity weak lensing convergence maps using generative adversarial networks. *Computational Astrophysics and Cosmology*, 6. 80

Neiswanger, W. and Ramdas, A. (2021). Uncertainty quantification using martingales for misspecified gaussian processes. *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 132:963–982. 5

Neyman, J. (1935). On the problem of confidence intervals. *Ann. Math. Statist.*, 6(3):111–116. 9

Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 236(767):333–380. 13, 93

Ong, V. M. H., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. (2017). Variational bayes with synthetic likelihood. *Statistics and Computing*, 28(4):971–988. 4

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv e-prints*, page arXiv:1609.03499. 4

Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *Proceedings of The 33rd International Conference on Machine Learning*, 48:1747–1756. 4

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4):502–518. 5

Papamakarios, G. and Murray, I. (2016). Fast ϵ-free inference of simulation models with bayesian conditional density estimation. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29, pages 1028–1036. Curran Associates, Inc. 4, 81, 94

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64. 4, 5

Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 2335–2344, USA. Curran Associates Inc. 5, 86

Papamakarios, G., Sterratt, D., and Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. 4, 94

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR. 118

Pasquet, J., Bertin, E., Treyer, M., Arnouts, S., and Fouchez, D. (2019). Photometric redshifts from SDSS images using a convolutional neural network. *Astronomy & Astrophysics*, 621:A26. 106

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 162, 163

Patil, P., Kuusela, M., and Hobbs, J. (2020). Objective frequentist uncertainty quantification for atmospheric CO_2 retrievals. *arXiv preprint 2007.14975*. 6, 123

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 118, 161

Pereira, C. A. d. B. and Stern, J. (1999). Evidence and credibility: full Bayesian significance test for precise hypotheses. *Entropy*, 1(4):99–110. 104

Peters, G. and Sisson, S. (2006). Bayesian inference, monte carlo sampling and operational risk. *Journal of Operational Risk*, 1. 3

Picchini, U., Simola, U., and Corander, J. (2020). Adaptive mcmc for synthetic likelihoods and correlated synthetic likelihoods. *arXiv preprint 2004.04558*. 4, 71

Pospisil, T. and Lee, A. B. (2018). RFCDE: Random Forests for Conditional Density Estimation. *arXiv preprint 1804.05753*. 97, 98

Pospisil, T. and Lee, A. B. (2019). (f)RFCDE: Random forests for conditional density estimation and functional data. *arXiv preprint 1906.07177*. 98

Prangle, D., Blum, M. G., Popovic, G., and Sisson, S. (2014). Diagnostic tools for approximate Bayesian computation using the coverage property. *Australian & New Zealand Journal of Statistics*, 56(4):309–329. 81

Pratt, J. W. (1961). Length of confidence intervals. *Journal of the American Statistical Association*, 56(295):549–567. 41

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11. 4, 71

Qian, X., Tan, A., Ling, J., Nakajima, Y., and Zhang, C. (2016). The gaussian CL_s method for searches of new physics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 827(35):63–78. 36

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. (2020). Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15. 4, 124

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. 6

Ratmann, O., Jørgensen, O., Hinkley, T., Stumpf, M., Richardson, S., and Wiuf, C. (2007). Using likelihood-free inference to compare evolutionary dynamics of the protein networks of h. pylori and p. falciparum. *PLOS Computational Biology*, 3(11):1–13. 3

Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J., and Póczos, B. (2017). Enabling dark energy science with deep generative models of galaxy images. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 1488–1494. AAAI Press. 80

Rizzo, M. (2021). R package "energy". *CRAN*. 162

Rodgers, C. D. (2000). *Inverse Methods for Atmospheric Sounding*. World Scientific. 6

Rolke, W. A., López, A. M., and Conrad, J. (2005). Limits and confidence intervals in the presence of nuisance parameters. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 551(2):493 – 503. 48

Rowe, B., Jarvis, M., Mandelbaum, R., Bernstein, G. M., Bosch, J., Simet, M., Meyers, J. E., Kacprzak, T., Nakajima, R., Zuntz, J., et al. (2015). GALSIM: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150. 59, 95, 109

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. K. (2019). Latent ordinary differential equations for irregularly-sampled time series. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 125

Schafer, C. M. and Stark, P. B. (2009). Constructing confidence regions of optimal expected size. *Journal of the American Statistical Association*, 104(487):1080–1089. 3

Schmidt, S. J., Malz, A. I., Soo, J. Y. H., Almosallam, I. A., Brescia, M., Cavuoti, S., Cohen-Tanugi, J., Connolly, A. J., DeRose, J., Freeman, P. E., and et al. (2020). Evaluation of probabilistic photometric redshift estimation approaches for the rubin observatory legacy survey of space and time (lsst). *Monthly Notices of the Royal Astronomical Society*. 24, 82, 101, 105

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681. 114

Sen, B., Walker, M., and Woodroofe, M. (2009). On the unified method with nuisance parameters. *Statistica Sinica*, 19(1):301–314. 35, 48

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC. 3

Sjöstrand, T., Edén, P., Friberg, C., Lönnblad, L., Miu, G., Mrenna, S., and Norrbin, E. (2001). High-energy-physics event generation with pythia 6.1. *Computer Physics Communications*, 135(2):238–259. 1

Smith, R. E., Peacock, J. A., Jenkins, A., White, S. D. M., Frenk, C. S., Pearce, F. R., Thomas, P. A., Efstathiou, G., and Couchman, H. M. P. (2003). Stable clustering, the halo model and non-linear cosmological power spectra. *Monthly Notices of the Royal Astronomical Society*, 341(4):1311–1332. 1

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2951–2959, Red Hook, NY, USA. Curran Associates Inc. 118

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc. 106

Stark, P. B. (1992). Inference in infinite-dimensional inverse problems: Discretization and duality. *Journal of Geophysical Research: Solid Earth*, 97(B10):14055–14082. 6

Stephens, M. A. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737. 110

Stevens, J. and Williams, M. (2013). uboost: a boosting method for producing uniform selection efficiencies from multivariate classifiers. *Journal of Instrumentation*, 8(12):P12013–P12013. 122

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, pages 1040–1053. 29

Stoye, M., Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. (2018). Likelihood-free inference with an improved cross-entropy estimator. *arXiv preprint 1808.00973*. 4

Stroud, P. D., Sydoriak, S. J., Riese, J. M., Smith, J. P., Mniszewski, S. M., and Romero, P. R. (2006). Semi-empirical power-law scaling of new infection rate to model epidemic dynamics with inhomogeneous mixing. *Mathematical Biosciences*, 203(2):301–318. 124

Stuart, A. M. (2010). Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559. 6

Sugiyama, M., Suzuki, T., Itoh, Y., Kanamori, T., and Kimura, M. (2011). Least-squares two-sample test. *Neural networks : the official journal of the International Neural Network Society*, 24:735–51. 77

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density Ratio Estimation in Machine Learning*. Cambridge University Press. 4

Szákely, G. J. and Rizzo, M. L. (2004). Testing for equal distributions in high dimensions. *InterStat*. 77, 83

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint 1804.06788*. 7, 24, 81

Tanabashi, M., Hagiwara, K., Hikasa, K., Nakamura, K., Sumino, Y., Takahashi, F., Tanaka, J., et al. (2018). Review of particle physics. *Phys. Rev. D*, 98:030001. 7, 9

Tang, Y. and Salakhutdinov, R. R. (2013). Learning stochastic feedforward neural networks. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 530–538. Curran Associates, Inc. 106

Thas, O. (2010). *Comparing distributions.* Springer. 77

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. (2021). Likelihood-free inference by ratio estimation. *Bayesian Anal.* Advance publication. 4

Thornton, S., Li, W., and ge Xie, M. (2017). An effective likelihood-free approximate computing method with statistical inferential guarantees. 5

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation. Revised and Extended from the 2004 French Original. Translated by Vladimir Zaiats.* Springer Series in Statistics. New York: Springer. 32

Uria, B., Côté, M.-A., Gregor, K., Murray, I., and Larochelle, H. (2016). Neural autoregressive distribution estimation. *Journal of Machine Learning Research*, 17(205):1–37. 4

Uria, B., Murray, I., and Larochelle, H. (2014). A deep and tractable density estimator. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 467–475, Bejing, China. PMLR. 4

van den Boom, W., Reeves, G., and Dunson, D. B. (2020). Approximating posteriors with high-dimensional nuisance parameters via integrated rotated gaussian approximation. *Biometrika.* 35

van Uitert, E., Joachimi, B., Joudaki, S., Amon, A., Heymans, C., Köhlinger, F., Asgari, M., Blake, C., Choi, A., Erben, T., Farrow, D. J., Harnois-Déraps, J., Hildebrandt, H., Hoekstra, H., Kitching, T. D., Klaes, D., Kuijken, K., Merten, J., Miller, L., Nakajima, R., Schneider, P., Valentijn, E., and Viola, M. (2018). KiDS+GAMA: cosmology constraints from a joint analysis of cosmic shear, galaxy-galaxy lensing, and angular clustering. *Monthly Notices of the Royal Astronomical Society*, 476(4):4662–4689. 94

Vanderplas, J., Connolly, A., Ivezić, Ž., and Gray, A. (2012). Introduction to astroML: Machine learning for astrophysics. In *Conference on Intelligent Data Understanding (CIDU)*, pages 47 –54. xxiv, 100

Vázquez, A., Flammini, A., Maritan, A., and A., V. (2003). Modeling of protein interaction networks. *ComPlexUs*, 1:38–44. 1

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. 70

Weinzierl, S. (2000). Introduction to monte carlo methods. 3, 121

Weir, I. S. (1997). Fully bayesian reconstructions from single-photon emission computed tomography data. *Journal of the American Statistical Association*, 92(437):49–60. 6

Wilkinson, R. (2014). Accelerating ABC methods using Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1015–1023. 4, 71

Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann. Math. Statist.*, 9(1):60–62. 93

Wiqvist, S., Frellsen, J., and Picchini, U. (2021). Sequential neural posterior and likelihood approximation. *arXiv preprint 2102.06522*. 4, 94

Wood, S. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466:1102–4. 3, 4, 71

Yang, Y., Bhattacharya, A., and Pati, D. (2017). Frequentist coverage and sup-norm convergence rate in gaussian process regression. *arXiv preprint 1708.04753*. 29

Young, G. A., Severini, T. A., Young, G. A., Smith, R., Smith, R. L., et al. (2005). *Essentials of statistical inference*, volume 16. Cambridge University Press. 121

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. (2021). Validating Conditional Density Models and Bayesian Inference Algorithms. *arXiv preprint 2102.10473*. 105

Zhu, Y., Shen, X., and Pan, W. (2020). On high-dimensional constrained maximum likelihood inference. *Journal of the American Statistical Association*, 115(529):217–230. 35

Appendix

Appendix A

Chapter 2 Appendix

A.1 Section 2.6 Proofs

Proof of Theorem 2.1. If we assume $|\Theta| < \infty$, then by the union bound and Assumption 2.1 we have that

$$\sup_{\theta\in\Theta_0}\sup_{t\in\mathbb{R}} |\widehat{F}_{B'}(t|\theta) - F(t|\theta)| \xrightarrow[B'\longrightarrow\infty]{\mathbb{P}} 0.$$

If we assume instead that $|\Theta|$ is a compact subset of $\mathbb{R}^d$, and the function $g_{B'}(\theta) = \sup_{t\in\mathbb{R}} |\widehat{F}_{B'}(t|\theta) - F(t|\theta)|$ is continuous in θ and strictly decreasing in $B^{'}$, then we can use Dini's theorem to obtain the above.

It follows that

$$\sup_{\theta\in\Theta_0} |\widehat{F}^{-1}_{B'}(\alpha|\theta) - F^{-1}(\alpha|\theta)| \xrightarrow[B'\longrightarrow\infty]{\mathbb{P}} 0.$$

The result follows from the fact that

$$0 \leq |C_{B,B'} - C^*_B| = |\sup_{\theta\in\Theta_0} \widehat{F}^{-1}_{B'}(\alpha|\theta) - \sup_{\theta\in\Theta_0} F^{-1}(\alpha|\theta)| \leq \sup_{\theta\in\Theta_0} |\widehat{F}^{-1}_{B'}(\alpha|\theta) - F^{-1}(\alpha|\theta)|,$$

and thus

$$|C_{B,B'} - C^*_B| \xrightarrow[B'\longrightarrow\infty]{\mathbb{P}} 0.$$

□

Lemma A.0.1. *If* $(\widehat{\mathbb{P}}(Y=1|\theta,\mathbf{X}))_{\theta\in\Theta} \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} (\mathbb{P}(Y=1|\theta,\mathbf{X}))_{\theta\in\Theta}$ *and* $|\Theta|<\infty$*, then*

$$\tau(\mathcal{D};\Theta_0) \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} \sup_{\theta_0\in\Theta_0}\inf_{\theta_1\in\Theta}\sum_{i=1}^{n}\log\left(\mathbb{OR}(\mathbf{X}_i^{obs};\theta_0,\theta_1)\right)$$

Proof. For every $\theta_0,\theta_1\in\Theta$, it follows directly from the properties of convergence in probability that

$$\sum_{i=1}^{n}\log\left(\widehat{\mathbb{OR}}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right) \xrightarrow[B\longrightarrow\infty]{\mathbb{P}} \sum_{i=1}^{n}\log\left(\mathbb{OR}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right)$$

The conclusion of the lemma follows from the continuous mapping theorem. $\square$

Proof of Theorem 2.2. Lemma A.0.1 implies that $\tau_B(\mathcal{D};\Theta_0)$ converges in distribution to $\sup_{\theta_0\in\Theta_0}\inf_{\theta_1\in\Theta}\sum_{i=1}^{n}\log\left(\mathbb{OR}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right)$. Now, from Slutsky's theorem,

$$\tau_B(\mathcal{D};\Theta_0)-\widehat{C}_B \xrightarrow[B\longrightarrow\infty]{\text{Dist}} \sup_{\theta_0\in\Theta_0}\inf_{\theta_1\in\Theta}\sum_{i=1}^{n}\log\left(\mathbb{OR}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right)-C^*.$$

It follows that

$$\begin{aligned}\mathbb{P}\left(\widehat{\phi}_{B,\widehat{C}_B}(\mathcal{D})=1|\theta\right) &= \mathbb{P}\left(\tau_B(\mathcal{D};\Theta_0)-\widehat{C}_B\le 0|\theta\right)\\ &\xrightarrow[B\longrightarrow\infty]{} \mathbb{P}\Big(\sup_{\theta_0\in\Theta_0}\inf_{\theta_1\in\Theta}\sum_{i=1}^{n}\log\left(\mathbb{OR}(\mathbf{X}_i^{\text{obs}};\theta_0,\theta_1)\right)-C^*\le 0|\theta\Big)\\ &= \mathbb{P}\left(\phi_{C^*}(\mathcal{D})=1|\theta\right),\end{aligned}$$

where the last equality follows from Proposition 1.

$\square$

Proof of Theorem 1. Assumption 1 implies that, for every D,

$$\begin{aligned}0\le|\widehat{p}(D;\Theta_0)-p(D;\Theta_0)| &= |\sup_{\theta\in\Theta_0}\widehat{p}(D;\theta)-\sup_{\theta\in\Theta_0}p(D;\theta)|\\ &\le \sup_{\theta\in\Theta_0}|\widehat{p}(D;\theta)-p(D;\theta)| \xrightarrow[B'\longrightarrow\infty]{\text{a.s.}} 0,\end{aligned}$$

and therefore $\widehat{p}(D;\Theta_0)$ converges almost surely to $p(D;\Theta_0)$. It follows that $\widehat{p}(\mathcal{D};\Theta_0)$ converges in distribution to $p(\mathcal{D};\Theta_0)$. Conclude that

$$\mathbb{P}_{\mathcal{D},\mathcal{T}'|\theta}(\widehat{p}(\mathcal{D};\Theta_0)\leq\alpha)=F_{\widehat{p}(\mathcal{D};\Theta_0)|\theta}(\alpha)\xrightarrow{B'\longrightarrow\infty}F_{p(\mathcal{D};\Theta_0)|\theta}(\alpha)=\mathbb{P}_{\mathcal{D}|\theta}(p(D;\Theta_0)\leq\alpha),$$

where F_Z denotes the cumulative distribution function of the random variable Z. □

Proof of Corollary 1. Fix $\theta\in\Theta$. Because F_θ is continuous, the definition of $p(\mathcal{D};\theta)$ implies that its distribution is uniform under the null. Thus $\mathbb{P}_{\mathcal{D}|\theta}\left(p(\mathcal{D};\theta)\leq\alpha\right)=\alpha$. Theorem 1 therefore implies that

$$\mathbb{P}_{\mathcal{D},\mathcal{T}'|\theta}(\widehat{p}(\mathcal{D};\theta)\leq\alpha)\xrightarrow{B'\longrightarrow\infty}\mathbb{P}_{\mathcal{D}|\theta}\left(p(\mathcal{D};\theta)\leq\alpha\right)=\alpha. \tag{A.1}$$

Now, for any $\theta\in\Theta_0$, uniformity of the p-value implies that

$$\begin{aligned}\mathbb{P}_{\mathcal{D}|\theta}(p(\mathcal{D};\Theta_0)\leq\alpha)&=\mathbb{P}_{\mathcal{D}|\theta}\left(\sup_{\theta_0\in\Theta_0}p(\mathcal{D};\theta_0)\leq\alpha\right)\leq\mathbb{P}_{\mathcal{D}|\theta}\left(p(\mathcal{D};\theta)\leq\alpha\right)\\&=\alpha.\end{aligned}$$

Conclude from Theorem 1 that

$$\mathbb{P}_{\mathcal{D},\mathcal{T}'|\theta}(\widehat{p}(\mathcal{D};\Theta_0)\leq\alpha)\xrightarrow{B'\longrightarrow\infty}\mathbb{P}_{\mathcal{D}|\theta}(p(\mathcal{D};\Theta_0)\leq\alpha)\leq\alpha. \tag{A.2}$$

The conclusion follows from putting together Equations A.1 and A.2. □

Proof of Theorem 2.

$$\begin{aligned}|\widehat{p}(D;\Theta_0)-p(D;\Theta_0)|&=|\sup_{\theta\in\Theta_0}\widehat{p}(D;\theta)-\sup_{\theta\in\Theta_0}p(D;\theta)|\\&\leq\sup_{\theta\in\Theta_0}|\widehat{p}(D;\theta)-p(D;\theta)|\\&=O_P\left(\left(\frac{1}{B'}\right)^r\right),\end{aligned}$$

where the last line follows from Assumption 2. □

Lemma 3. *Under Assumption 4,* $\int(\mathbb{O}(\mathbf{x};\theta_0)-\widehat{\mathbb{O}}(\mathbf{x};\theta_0))^2dG(\mathbf{x})\leq\frac{M'}{m'}L(\widehat{\mathbb{O}},\mathbb{O})$.

Proof. Let h be as in Assumption 4. Notice that

$$\begin{aligned}\int (\mathbb{O}(\mathbf{x};\theta_0) - \widehat{\mathbb{O}}(\mathbf{x};\theta_0))^2 dG(\mathbf{x}) &\leq \sup_{\theta\in\Theta} \int (\mathbb{O}(\mathbf{x};\theta) - \widehat{\mathbb{O}}(\mathbf{x};\theta))^2 dG(\mathbf{x}) \\ &\leq M' \leq \frac{M'}{m'} \int h(\theta) d\pi(\theta) \\ &= \frac{M'}{m'} L(\widehat{\mathbb{O}}, \mathbb{O}),\end{aligned}$$

which concludes the proof.

□

Lemma 4. *Under Assumptions 3 and 4, there exists $K > 0$ such that*

$$\mathbb{E}_{\mathcal{D}|\theta,T_B}\left[|\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|\right] \leq K\sqrt{L(\widehat{\mathbb{O}}, \mathbb{O})}.$$

Proof. For every $\theta \in \Theta$

$$\begin{aligned}\mathbb{E}^2_{\mathcal{D}|\theta,T_B}[|\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|] &= \left(\int |\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|\ dF(\mathbf{x}|\theta)\right)^2 \\ &= \left(\int |\mathbb{O}(\mathbf{x};\theta_0) - \widehat{\mathbb{O}}(\mathbf{x};\theta_0)|\ dF(\mathbf{x}|\theta)\right)^2 \\ &= \left(\int |\mathbb{O}(\mathbf{x};\theta_0) - \widehat{\mathbb{O}}(\mathbf{x};\theta_0)| \mathbb{O}(\mathbf{x};\theta) dG(\mathbf{x})\right)^2 \\ &\leq \left(\int (\mathbb{O}(\mathbf{x};\theta_0) - \widehat{\mathbb{O}}(\mathbf{x};\theta_0)^2 dG(\mathbf{x})\right)\left(\int \mathbb{O}^2(\mathbf{x};\theta) dG(\mathbf{x})\right),\end{aligned}$$

where the last inequality follows from Cauchy-Schwarz. Assumption 3 implies that

$$\int \mathbb{O}^2(\mathbf{x};\theta) dG(\mathbf{x}) \leq M^2,$$

from which we conclude that

$$\mathbb{E}^2_{\mathcal{D}|\theta,T_B}[|\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|] \leq M^2 \int (\mathbb{O}(\mathbf{x};\theta_0) - \widehat{\mathbb{O}}(\mathbf{x};\theta_0))^2 dG(\mathbf{x}).$$

Conclude from Lemma 3 that

$$\mathbb{E}^2_{\mathcal{D}|\theta,T_B}[|\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|] \leq K^2 \cdot L(\widehat{\mathbb{O}}, \mathbb{O}),$$

where $K = M\sqrt{\frac{M'}{m'}}$. $\square$

Lemma 5. *Under Assumptions 3-7, there exists $C > 0$ such that*

$$\mathbb{E}_{\mathcal{D},\mathcal{T}_B|\theta}\left[|\tau(\mathcal{D};\theta_0) - \widehat{\tau}_B(\mathcal{D};\theta_0)|\right] \leq CB^{-\alpha/(2(\alpha+d))}.$$

Proof. Let $\widehat{p} = \widehat{\mathbb{P}}(Y = 1|\mathbf{x},\theta)$ and $p = \mathbb{P}(Y = 1|\mathbf{x},\theta)$ be the probabilistic classifier and true classification function, respectively, on the training sample D_B. Let $h(y) = \frac{y}{1-y}$ for $0 < y < 1$. A Taylor expansion of h implies that

$$(h(\widehat{p}) - h(p))^2 = (h(p) + R_1(\widehat{p}) - h(p))^2 = R_1(\widehat{p})^2,$$

where $R_1(\widehat{p}) = h'(\xi)(\widehat{p} - p)$ for some ξ between p and $\widehat{p}$. Also note that due to Assumption 3,

$$\exists a > 0 \text{ s.t. } p, \widehat{p} > a, \ \forall x \in \mathcal{X}, \theta \in \Theta.$$

Thus,

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}_B}\left[\int (h(\widehat{p}) - h(p))^2 \, dG(\mathbf{x})d\pi(\theta)\right] &= \mathbb{E}_{\mathcal{T}_B}\left[\int \frac{1}{(1-\xi)^4}(\widehat{p} - p)^2 \, dG(\mathbf{x})d\pi(\theta)\right] \\
&\leq \frac{1}{(1-a)^4}\mathbb{E}_{\mathcal{T}_B}\left[\int (\widehat{p} - p)^2 \, dG(\mathbf{x})d\pi(\theta)\right] \\
&= \frac{1}{(1-a)^4}\mathbb{E}_{\mathcal{T}_B}\left[\int \left(\widehat{\mathbb{P}}(Y = 1|\mathbf{x},\theta) - \mathbb{P}(Y = 1|\mathbf{x},\theta)\right)^2 h'(\mathbf{x},\theta)dH(\mathbf{x},\theta)\right] \\
&\leq \frac{\gamma}{(1-a)^4}\mathbb{E}_{\mathcal{T}_B}\left[\int \left(\widehat{\mathbb{P}}(Y = 1|\mathbf{x},\theta) - \mathbb{P}(Y = 1|\mathbf{x},\theta)\right)^2 dH(\mathbf{x},\theta)\right] \\
&= O\left(B^{-\alpha/(\alpha+d)}\right)
\end{aligned}$$

It follows that

$$\begin{aligned}\mathbb{E}_{\mathcal{D},\mathcal{T}_B|\theta}\left[|\tau(\mathcal{D};\theta_0)-\widehat{\tau}_B(\mathcal{D};\theta_0)|\right] &= \mathbb{E}_{\mathcal{T}_B}\left[\mathbb{E}_{\mathcal{D}|\theta,\mathcal{T}_B}\left[|\tau(\mathcal{D};\theta_0)-\widehat{\tau}_B(\mathcal{D};\theta_0)|\right]\right]\\ &\leq \mathbb{E}_{\mathcal{T}_B}\left[K\sqrt{L(\widehat{\mathbb{O}},\mathbb{O})}\right]\\ &\leq K\sqrt{\mathbb{E}_{\mathcal{T}_B}\left[L(\widehat{\mathbb{O}},\mathbb{O})\right]}\\ &= K\sqrt{\mathbb{E}_{\mathcal{T}_B}\left[\int\left(h(\widehat{p})-h(p)\right)^2 dG(\mathbf{x})d\pi(\theta)\right]}\\ &= O\left(B^{-\alpha/(2(\alpha+d))}\right),\end{aligned}$$

where the second inequality follows from Lemma 4. □

Proof of Theorem 3. It follows from Markov's inequality that with probability at least $1-\epsilon$, $\mathcal{D}$ is such that

$$|\tau(\mathcal{D};\theta_0)-\widehat{\tau}(\mathcal{D};\theta_0)| \leq \frac{K\cdot\sqrt{L(\widehat{\mathbb{O}},\mathbb{O})}}{\epsilon} \tag{A.3}$$

Now we upper bound $\mathbb{P}_\theta(\phi_\tau(\mathcal{D})\neq\phi_\tau(\mathcal{D}))$. Define A as the event that Eq. A.3 happens. Then:

$$\begin{aligned}\mathbb{P}_{\mathcal{D}|\theta,D_B}(\phi_\tau(\mathcal{D})\neq\phi_{\widehat{\tau}}(\mathcal{D})) &\leq \mathbb{P}_{\mathcal{D}|\theta,D_B}(\phi_\tau(\mathcal{D})\neq\phi_{\widehat{\tau}}(\mathcal{D}),A)+\mathbb{P}_\theta(A^c)\\ &\leq \mathbb{P}_{\mathcal{D}|\theta,D_B}\left(\mathbb{I}\left(\tau(\mathcal{D};\theta_0)<c\right)\neq\mathbb{I}\left(\widehat{\tau}(\mathcal{D};\theta_0)<c\right),A\right)+\epsilon\\ &\leq \mathbb{P}_{\mathcal{D}|\theta,D_B}\left(c-\frac{K\cdot\sqrt{L(\widehat{\mathbb{O}},\mathbb{O})}}{\epsilon}<\tau(\mathcal{D};\theta_0)<c+\frac{K\cdot\sqrt{L(\widehat{\mathbb{O}},\mathbb{O})}}{\epsilon}\right)+\epsilon\end{aligned}$$

Assumption 5 then implies that

$$\mathbb{P}_{\mathcal{D}|\theta,D_B}(\phi_\tau(\mathcal{D})\neq\phi_{\widehat{\tau}}(\mathcal{D})) \leq \frac{K'\cdot\sqrt{L(\widehat{\mathbb{O}},\mathbb{O})}}{\epsilon}+\epsilon$$

where $K'=2KC_L$, which concludes the proof. □

Proof of Theorem 4. It follows from Markov's inequality that with probability at least $1-\epsilon$, $\mathcal{D}$ is such that

$$|\tau(\mathcal{D};\theta_0) - \widehat{\tau}(\mathcal{D};\theta_0)| \leq \frac{CB^{-\alpha/(2(\alpha+d))}}{\epsilon} \tag{A.4}$$

Following the same reasoning as for Theorem 3, we obtain that

$$\mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_\tau(\mathcal{D}) \neq \phi_{\widehat{\tau}}(\mathcal{D})) \leq \frac{K''B^{-\alpha/(2(\alpha+d))}}{\epsilon} + \epsilon$$

where $K'' = 2CC_L$. Notice that taking $\epsilon^* = \sqrt{K''}B^{-\alpha/(4(\alpha+d))}$ optimizes the bound and gives the result. $\square$

Proof of Corollary 2. The result follows from noticing that

$$\begin{aligned}\mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_{\widehat{\tau}_B}(\mathcal{D}) = 1) &\geq \mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_\tau(\mathcal{D}) = 1) - \mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_\tau(\mathcal{D}) \neq \phi_{\widehat{\tau}_B}(\mathcal{D})) \\ &\geq \mathbb{P}_{\mathcal{D},\mathcal{T}_B|\theta}(\phi_\tau(\mathcal{D}) = 1) - 2\sqrt{K''}B^{-\alpha/(4(\alpha+d))},\end{aligned}$$

where the last inequality follows from Theorem 4. $\square$

Appendix B

Source Code and Implementation Details

B.1 Chapter 2

For the examples in Section 2.9.1, we use the `sklearn` ecosystem Pedregosa et al. (2011) implementation of the following probabilistic classifiers:

- multi-layer perceptron (MLP) with default parameters, but no L^2 regularization ($\alpha = 0$);
- quadratic discriminant analysis (QDA) with default parameters;
- nearest neighbors (NN) classifier, with number of neighbors equal to the rounded square root of the number of data points available (as per Duda et al. (2001)).

For the examples in Sections 2.9.3 and 2.9.4, we construct confidence sets using the `ACORE` test statistic. For learning the odds ratio, we compared the following classifiers:

- logistic regression,
- quadratic disciminінant analysis (QDA) classifier,
- nearest neighbor classifier,
- gradient boosted trees using $\{100, 500, 1000\}$ trees with maximum depth $\{3, 5, 10\}$,

- Gaussian process classifiers[*] with radial basis functions kernels with variance $\{1, .5, .1\}$,
- feed-forward deep neural networks, with $2, ..., 6$ deep layers, number of neurons between $2^{\{4,...,10\}}$ and either ReLu or hyperbolic tangent activations.

For estimating the critical values, we considered the following quantile regression algorithms:

- gradient boosted trees using $\{100, 250, 500\}$ trees with maximum depth $\{3, 5, 10\}$,
- random forest quantile regression with $\{100, 250, 500\}$ trees,
- deep quantile regression with $\{2, 3\}$ deep layers, $2^{\{4,..,6\}}$ neurons and ReLu activations (using the `PyTorch` implementation Paszke et al. (2019)).

All computations were performed on 8-Core Intel Xeon CPUs X5680 at 3.33GHz. A `Python` implementation of all experiments and construction of `ACORE` and `BFF` confidence sets can be found on `Github` at `Mr8ND/ACORE-LFI`.

B.2 Chapter 3

For Section 3.2, MMD and Energy test statistics are implemented in `R`, using the `energy` (Rizzo, 2021) and `kernlab` Karatzoglou et al. (2004) packages respectively. Random forest and nearest neighbors algorithms for Section 3.1.3 are also implemented in `R`. The random forest algorithm is vanilla implementation from the `randomForest` package(Liaw and Wiener, 2002), while the nearest neighbor algorithm is taken from the `caret` package (Kuhn, 2008), choosing the number of nearest neighbors as the square root of the total number of points as in Lall and Sharma (1996).

The conditional MAF in Section 3.3 is implemented in `Python3` in `pyTorch` (Paszke et al., 2019). At both $n_{\text{train}} = 200$ and $n_{\text{train}} = 500$ we used 10% of the training data as validation. During training we assessed validation loss and we stopped the training early if the validation loss was not improving for 30 epochs. We explored architectures with

[*]GP classifiers were used only with sample sizes B below $10,000$, as the matrix inversion quickly becomes computationally infeasible for larger values of B.

$\{5, 10, 15, 20\}$ autoregressive layers and $2^{\{4,..,10\}}$ hidden units, with the best performing having 10 autoregressive layers and either 512 or 1024 hidden units.

A barebone implementation of the two sample regression test for both `R` and `Python` can be found on `Github` at `Mr8ND/Emulator-Validation-LFI`

B.3 Chapter 4

`DeepCDE` has implementations for both `Tensorflow` (Abadi et al., 2015) and `Pytorch` (Paszke et al., 2019), two of the most widely used deep learning frameworks. Implementations can be found at `Mr8ND/DeepCDE`.