Likelihood-free inference with emulator networks

Jan-Matthis Lueckmann[1,2] JAN-MATTHIS.LUECKMANN@CAESAR.DE
Giacomo Bassetto[1,2] GIACOMO.BASSETTO@CAESAR.DE
Theofanis Karaletsos[3] THEOFANIS@UBER.COM
Jakob H. Macke[1,2,4] MACKE@TUM.DE

Abstract

Approximate Bayesian Computation (ABC) provides methods for Bayesian inference in simulation-based models which do not permit tractable likelihoods. We present a new ABC method which uses probabilistic neural *emulator* networks to learn synthetic likelihoods on simulated data – both 'local' emulators which approximate the likelihood for specific observed data, as well as 'global' ones which are applicable to a range of data. Simulations are chosen adaptively using an acquisition function which takes into account uncertainty about either the posterior distribution of interest, or the parameters of the emulator. Our approach does not rely on user-defined rejection thresholds or distance functions. We illustrate inference with emulator networks on synthetic examples and on a biophysical neuron model, and show that emulators allow accurate and efficient inference even on problems which are challenging for conventional ABC approaches.

1. Introduction

Many areas of science and engineering make extensive use of complex, stochastic, numerical simulations to describe the structure and dynamics of the processes being investigated (Karabatsos and Leisen, 2017). A key challenge in simulation-based science is linking simulation models to empirical data: Bayesian inference provides a general and powerful framework for identifying the set of parameters which are consistent both with empirical data and prior knowledge. One of the key quantities required for statistical inference, the likelihood of observed data given parameters, $\mathcal{L}(\boldsymbol{\theta}) = p(\mathbf{x}_o|\boldsymbol{\theta})$, is typically intractable for simulation-based models, rendering conventional statistical approaches inapplicable.

Approximate Bayesian Computation (ABC) aims to close this gap (Beaumont et al., 2002), but classical algorithms (Pritchard et al., 1999; Marjoram et al., 2003) scale poorly to high-dimensional non-Gaussian data, and require *ad-hoc* choices (i.e., rejection thresholds, distance functions and summary statistics) which can significantly affect both computational efficiency and accuracy. In *synthetic likelihood* approaches to ABC (Wood, 2010; Ong et al., 2016; Price et al., 2018), one instead uses density estimation to approximate the likelihood $p(s(\mathbf{x}_o)|\boldsymbol{\theta})$ on summary statistics $s(\cdot)$ of simulated data. A recent proposal by Järvenpää et al. (2017), Gutmann and Corander (2016) uses a Gaussian process ($\mathcal{GP}$) to approximate the distribution of the discrepancy $d(s(\mathbf{x}), s(\mathbf{x}_o))$ as a function of $\boldsymbol{\theta}$, and Bayesian Optimization

[1] Computational Neuroengineering, Technical University of Munich, Germany
[2] Research Center caesar, an associate of the Max Planck Society, Bonn, Germany
[3] Uber AI Labs, Uber Technologies, Inc., San Francisco, CA
[4] Part of this work was done while J.H.M was at the Centre for Cognitive Science, Technische Universität Darmstadt, Germany

to propose new parameters. While this approach can be very effective even with a small number of simulations, it still requires summary statistics, choice of a distance function $d(\cdot,\cdot)$, and relies on assuming a homeoscedastic $\mathcal{GP}$. Appendix A discusses additional related work.

The goal of this paper is to scale synthetic-likelihood method to multivariate and (potentially) non Gaussian, heteroscedastic data. We use neural-network based conditional density estimators (which we call 'emulator networks', inspired by classical work on emulation methods; Kennedy and O'Hagan, 2002), to develop likelihood-free inference algorithms which are efficient, flexible, and scale to high-dimensional observations. Our approach does not require the user to specify rejection thresholds or distance functions, or to restrict oneself to a small number of summary statistics.

2. Likelihood-free inference with emulator networks

Our goal is to obtain an approximation to the true posterior $p(\boldsymbol{\theta}|\mathbf{x}_o)$ of a black-box simulator model, i.e. models from which we can generate samples $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$, but for which we cannot evaluate likelihoods $\mathcal{L}(\boldsymbol{\theta})$. Core to our approach is an emulator $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$, a conditional density estimator with parameters $\boldsymbol{\phi}$ that approximates the simulator $p(\mathbf{x}|\boldsymbol{\theta})$. Having collected an initial simulated dataset $\mathcal{D}$, e.g. by repeatedly drawing from the prior $p(\boldsymbol{\theta})$ and simulating data, the emulator is trained. We actively select new locations $\boldsymbol{\theta}^*$ for which to simulate new data points $\mathcal{D}^* = \{(\boldsymbol{\theta}^*, \mathbf{x}^*)\}$ to keep the number of calls to the (potentially computationally expensive) simulator low. $\mathcal{D}^*$ is appended to the dataset, the emulator is updated, and the active learning loop repeats.

The emulator defines a synthetic likelihood function $\hat{\mathcal{L}}(\boldsymbol{\theta};\boldsymbol{\phi}) = q(\mathbf{x} = \mathbf{x}_o|\boldsymbol{\theta};\boldsymbol{\phi})$ that we use to find an approximate posterior, which proportional to $\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o) := \hat{\mathcal{L}}(\boldsymbol{\theta})p(\boldsymbol{\theta})$. We draw samples from $\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o)$ using Hamiltonian Monte Carlo (Neal, 2010). We use probabilistic neural networks as emulators, i.e. we represent uncertainty about the parameters $\boldsymbol{\phi}$ of $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$. We then use these uncertainties to guide the selection of sample points using active learning. In our experiments, we use deep ensembles (Lakshminarayanan et al., 2016), as we found them to combine simplicity with good empirical performance. The emulator is a mixture distribution over M ensemble networks: $\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\big[q(\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi})\big] \approx \frac{1}{M}\sum_{m=1}^{M} q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi}_m)$.

We use active learning to selectively acquire new samples. We distinguish between two scenarios: In the first, we have particular observed data $\mathbf{x}_o$ available, and train a *local emulator* for inference. Approximating the likelihood near $\mathbf{x}_o$ is an easier goal than globally approximating the simulator for all possible $\mathbf{x}$. However, this approach requires learning a new emulator for each new observed data $\mathbf{x}_o$. Therefore, we also consider a second scenario, in which we learn a *global emulator* that is valid for a range of data. Learning a global emulator is more challenging and may potentially require more flexible density estimators. However, once the emulator is learned, we can readily approximate the likelihood for *any* $\mathbf{x}_o$.

Acquisitions for local emulator learning: As we are interested in increasing our certainty about the posterior, we target its variance, $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o,\boldsymbol{\phi})]$, where $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}$ denotes that we take the variance with respect to the posterior over network weights given data $\mathcal{D}$. We refer to this rule as *MaxVar* (Järvenpää et al., 2017, details in Appendix B).

Acquisitions for global emulator learning: We use a rule based on mutual information from the active learning literature (Houlsby et al., 2011; Gal et al., 2017; Depeweg et al., 2017), selecting $\boldsymbol{\theta}^*$ as $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{I}[\mathbf{x},\boldsymbol{\phi}|\boldsymbol{\theta},\mathcal{D}]$. We will refer to this rule as the maximum mutual information rule, *MaxMI* (details in Appendix C).

3. Results

We demonstrate likelihood-free inference with emulator networks on three examples: i) we show that emulators are competitive with state-of-the-art on an example with Gaussian observations; ii) we demonstrate the ability of emulators to work with high-dimensional observations while learning to amortize the simulator; iii) we show an application from neuroscience, and infer the posterior over parameters of a biophysical neuron model.

i) Low-dimensional example: Simulator with Gaussian observations

We aimed to estimate data of a non-linear mapping from parameters to data, corrupted by additive Gaussian observation noise (Fig. 1a; details in Appendix D). We compare our emulator-method to BOLFI, a state-of-the-art for simulation-efficient inference which has been shown to be substantially more efficient than classical methods, including rejection-ABC, MCMC-ABC, and SMC-ABC (Gutmann and Corander, 2016). Both BOLFI variants (*ExpIntVar* and *MaxVar*) exhibit similar performance to emulators, but require higher number of simulations (comparison in Fig. 1b–c).

ii) High-dimensional observations: Inferring the location and contrast of a blob

We show that our method can be applied to estimation problems with high-dimensional observations without having to resort to using summary statistics. We model the rendering of a blob on a 2D image, and learn a global emulator for the forward model. The forward model takes as inputs three parameters (x_{off}, y_{off}, γ) – which encode horizontal and vertical displacement, and contrast of the blob – and returns binomial per-pixel activation probabilities (Fig. 2a; details in Appendix E).

Using the *MaxMI* rule to acquire new test points in parameters space results in faster learning of the emulator, compared to uniform random acquisitions. Eventually, both rules converge towards the log-likelihood of the held-out test set, indicating successful global emulation of the forward model (Fig. 2b).

iii) Scientific application: Hodgkin-Huxley model

As an example of a scientific application, we use the Hodgkin-Huxley model (Hodgkin and Huxley, 1952) which describes the evolution of membrane potential in neurons (Fig. 3a; details in Appendix F). Fitting single- and multi-compartment Hodgkin-Huxley models to neurophysiological data is a central problem in neuroscience, and typically addressed using heuristic, non-Bayesian methods (Druckmann et al., 2007; Van Geit et al., 2016). In contrast to the previous examples, we infer the posterior for summary features (number of spikes), as they are of direct interest in this application (posteriors and predictive checks in Fig. 3b).

4. Discussion

Numerical simulations make it possible to model complex phenomena from first principles, and are indispensable tools in many fields in engineering and science. Our Bayesian methodology based on emulators provides a fast, effective surrogate model for the intractable likelihood implied by the simulator, and the active-learning based rules lead to bounded-rational decisions about which simulations to run.

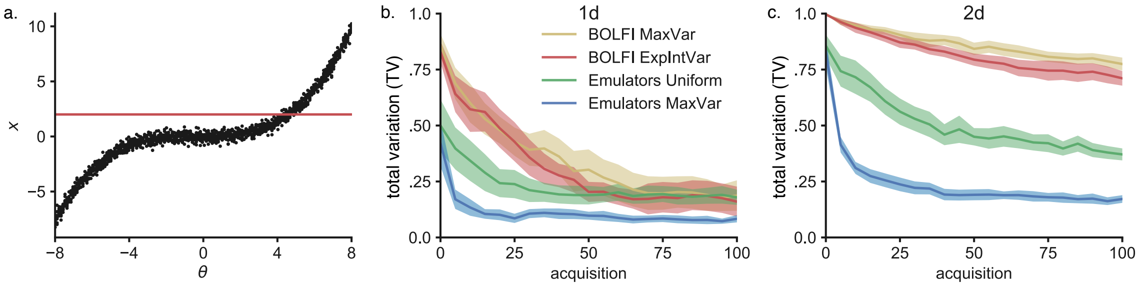

Figure 1: **Inference on simulator with Gaussian noise**. **a.** Data is generated from $\mathbf{x} \sim \mathcal{N}(\mathbf{x}|f(\boldsymbol{\theta}), \boldsymbol{\Sigma})$ with cubic non-linearity; posterior inference given $\mathbf{x}_o = 2$ (red). **b.** In 1-D, emulator-based inference with the *MaxVar* acquisitions leads to faster convergence to the true posterior than uniform sampling, or BOLFI. TV between true and approximate posterior. Lines are means and SEMs from 20 runs. See Appendix G for convergence of BOLFI. **c.** Same problem, but $\mathbf{x}$ and $\boldsymbol{\theta} \in \mathbb{R}^2$ and the non-linearity is applied point-wise.

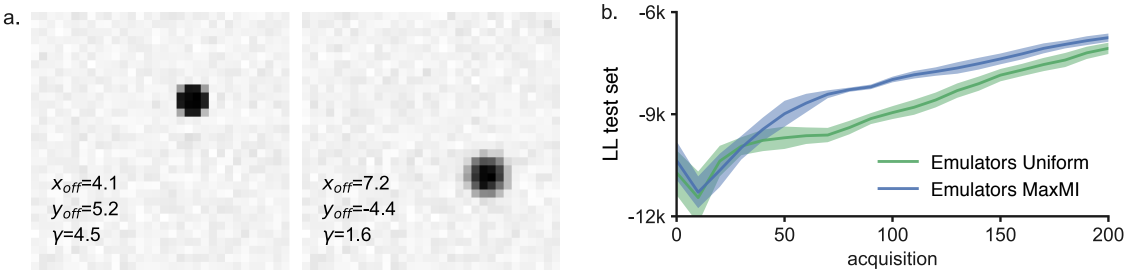

Figure 2: **Inferring location and scale of a blob**. **a.** Two sample images from the generative model. Parameters are the spatial position and the contrast of the blob. **b.** Acquiring samples using the *MaxMI* rule yield to faster emulator learning than samples acquired uniformly in the parameter space. Performances are reported as log-likelihood of held-out test data.

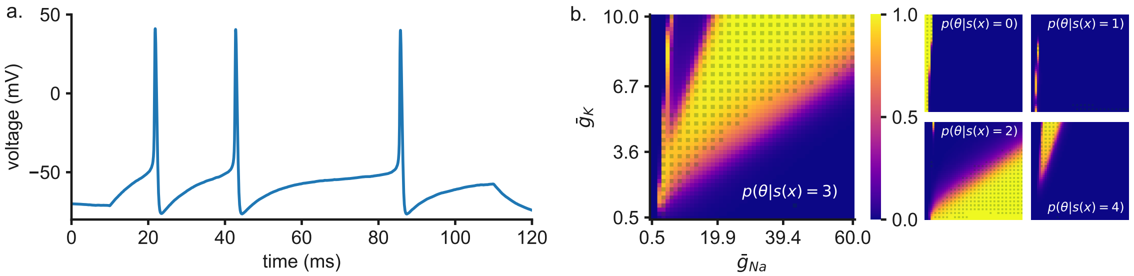

Figure 3: **Hodgkin-Huxley model**. **a.** Example trace from differential equations describing the model. **b.** Posterior inferred for number of spikes as a function of two biophysical parameters. Panels show posteriors for a given number of spikes. As a posterior predictive check, we overlay transparent squares on top of the posteriors where simulations produced the given number of spikes (and no squares otherwise). See Appendix H for details on acquisitions during active learning.

Acknowledgements

We thank Marcel Nonnenmacher and Pedro J. Gonçalves for discussions, and help with simulation of the Hodgkin-Huxley model. We thank David Greenberg and all members of the Neural Systems Analysis group for comments on the manuscript.

This work was supported by BMBF (FKZ 01IS18052 A-D, Project ADIMEM) and DFG (SFB 1089, SFB 1233, and SPP 2041) grants to JHM, as well as by the caesar foundation.

References

M Beaumont, W Zhang, and D J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4), 2002.

S Depeweg, J M Hernández-Lobato, F Doshi-Velez, and S Udluft. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *arXiv:1710.07283*, 2017.

S Druckmann, Y Banitt, A Gidon, F Schürmann, H Markram, and I Segev. A novel multiple objective optimization framework for constraining conductance-based neuron models by experimental data. *Frontiers in Neuroscience*, 1, 2007.

Y Gal, R Islam, and Z Ghahramani. Deep bayesian active learning with image data. *arXiv:1703.02910*, 2017.

M U Gutmann and J Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. *Journal of Machine Learning Research*, 17(125), 2016.

A L Hodgkin and A F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), 1952.

N Houlsby, F Huszar, Z Ghahramani, and M Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

M Järvenpää, M U Gutmann, A Pleska, Vehtari, A, and P Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *arXiv:1704.00520v2*, 2017.

G Karabatsos and F Leisen. An Approximate Likelihood Perspective on ABC Methods. *arXiv:1708.05341*, 2017.

M C Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3), 2002.

B Lakshminarayanan, A Pritzel, and C Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *arXiv:1612.01474*, 2016.

P Marjoram, J Molitor, V Plagnol, and S Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26), 2003.

R M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

V M H Ong, D J Nott, M Tran, S A Sisson, and C C Drovandi. Variational bayes with synthetic likelihood. *arXiv:1608.03069*, 2016.

L F Price, C C Drovandi, A Lee, and D J Nott. Bayesian Synthetic Likelihood. *Journal of Computational and Graphical Statistics*, 27(1), 2018.

J K Pritchard, M T Seielstad, A Perez-Lezaun, and M W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Mol Biol Evol*, 16(12), 1999.

W Van Geit, M Gevaert, G Chindemi, C Rössert, JD Courcol, E B Muller, F Schürmann, I Segev, and H Markram. Bluepyopt: Leveraging open source software and cloud infrastructure to optimise model parameters in neuroscience. *Frontiers in Neuroinformatics*, 10, 2016.

S N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310), 2010.

Appendix

Appendix A. Additional related work

Our approach uses density estimation to approximate the likelihood. A complementary use of density-estimation in ABC is to directly target the posterior distribution (Papamakarios and Murray, 2017; Lueckmann et al., 2018b; Le et al., 2017; Izbicki et al., 2018). This approach can be very useful – however, one advantage of likelihood-based approaches is that they allow one to apply the same synthetic likelihood to multiple priors (without having to retrain), or to pool information from multiple observations (by multiplying the corresponding synthetic likelihoods). More technically, posterior density estimation gives less flexibility in proposing samples – in order to yield the correct posterior, samples have to be drawn from the prior, or approaches such as importance-weighting (Lueckmann et al., 2018b) or post-hoc corrections (Papamakarios and Murray, 2017) have to be applied.

Papamakarios et al. (2018), concurrently and independently to our approach (Lueckmann et al., 2018a, an earlier preprint version of this work), proposed learning synthetic likelihoods using neural density estimators for likelihood-free inference: They use Masked Autoregressive Flows as synthetic likelihoods and report state-of-the-art performance compared to methods that directly target the posterior. Like our approach, the density estimator is trained on sequentially chosen simulations. Rather than using acquisition functions that take into account uncertainty to guide sampling, they draw samples from the current estimate of the posterior. Their approach corresponds to an alternative way of learning a local emulator.

The recent workshop paper of Durkan et al. (2018) compares Papamakarios et al. (2018) and our approach on three toy problems learning local emulators. On these toy-problems, both methods are similarly efficient (and more efficient than methods directly targeting the posterior), however, the wallclock time of our method is substantially higher, because of the additional cost of evaluating the acquisition function. Thus, whether this additional cost is warranted on a given problem will depend both on any additional gain brought about by the active selection of samples, as well as the cost of the simulator. For expensive simulation costs, additional computational budget should be spent to carefully decide for which parameters to simulate.

References

C Durkan, G Papamakarios, and I Murray. Sequential neural methods for likelihood-free inference. *arXiv:1811.08723*, 2018.

R Izbicki, A B Lee, and T Pospisil. Abc-cde: Towards approximate bayesian computation with complex high-dimensional data and limited simulations. *arXiv:1805.05480*, 2018.

T A Le, A G Baydin, R Zinkov, and F D Wood. Using synthetic data to train neural networks is model-based reasoning. *arXiv:1703.00868*, 2017.

JM Lueckmann, G Bassetto, T Karaletsos, and J H Macke. Likelihood-free inference with emulator networks. *arXiv:1805.09294v1*, 2018a.

JM Lueckmann, P J Goncalves, G Bassetto, K Öcal, M Nonnenmacher, and J H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2018b.

G Papamakarios and I Murray. Fast epsilon-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, 2017.

G Papamakarios, D C Sterratt, and I Murray. Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. *arXiv:1805.07226*, 2018.

Appendix B. Acquisition rule for local emulator learning

With given $\mathbf{x}_o$, we want to learn a local emulator that allows us to derive a good approximation to the (unnormalized) posterior

$$\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o) \propto \mathbb{E}_{\phi|\mathcal{D}}\big[q(\mathbf{x} = \mathbf{x}_o|\boldsymbol{\theta};\boldsymbol{\phi})\big]p(\boldsymbol{\theta}). \tag{1}$$

As we are interested in increasing our certainty about the posterior, we target its variance, $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o, \boldsymbol{\phi})]$, where $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}$ denotes that we take the variance with respect to the posterior over network weights given data $\mathcal{D}$. Thus, we use an acquisition rule which targets the region of maximum variance in the predicted (unnormalized) posterior,

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\tilde{p}(\boldsymbol{\theta}|\mathbf{x}_o, \boldsymbol{\phi})] \\ &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta})^2\ \mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\hat{\mathcal{L}}(\boldsymbol{\theta})] \\ &= \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \log \sqrt{\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}[\hat{\mathcal{L}}(\boldsymbol{\theta})]}.\end{aligned} \tag{2}$$

For all practical purposes, we approximate $\mathbb{V}_{\boldsymbol{\phi}|\mathcal{D}}$ with the sample variance taken across $\boldsymbol{\phi}_m$ drawn from the posterior over networks. We refer to this rule as the *MaxVar* rule (Järvenpää et al., 2017).

In practice, we optimize this acquisition rule by using gradient descent, making use of automatic differentiation to take gradients with respect to $\boldsymbol{\theta}$ through the synthetic likelihood function specified by the emulator.

References

M Järvenpää, M U Gutmann, A Pleska, Vehtari, A, and P Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *arXiv:1704.00520v2*, 2017.

Appendix C. Acquisition rule for global emulator learning

A global emulator may be used to do inference once $\mathbf{x}_o$ becomes available. Here, the goal for active learning is to bring the emulator $q(\mathbf{x}|\boldsymbol{\theta};\boldsymbol{\phi})$ close to the simulator $p(\mathbf{x}|\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$s using as few runs of the simulator as possible. We use a rule based on information theory from the active learning literature (Houlsby et al., 2011; Gal et al., 2017; Depeweg et al., 2017). We refer to the rule

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} \mathbb{I}[\mathbf{x},\boldsymbol{\phi}|\boldsymbol{\theta},\mathcal{D}] \\ &= \arg\max_{\boldsymbol{\theta}} \underbrace{\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\mathcal{D}]}_{\text{entropy}} - \underbrace{\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\left[\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}]\right]}_{\text{expected conditional entropy}}\end{aligned} \tag{3}$$

as the maximum mutual information rule (*MaxMI*).

The first term is the entropy of the data under the posterior-predictive distribution implied by the emulator:

$$\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\mathcal{D}] = -\int \hat{p}(\mathbf{x}|\boldsymbol{\theta},\mathcal{D}) \ln \hat{p}(\mathbf{x}|\boldsymbol{\theta},\mathcal{D})\mathrm{d}\mathbf{x}, \tag{4}$$

where $\hat{p}(\mathbf{x}|\boldsymbol{\theta},\mathcal{D})$ is obtained by marginalizing out the emulator's parameters w.r.t. $p(\boldsymbol{\phi}|\mathcal{D})$:

$$\hat{p}(\mathbf{x}|\boldsymbol{\theta},\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi})p(\boldsymbol{\phi}|\mathcal{D})\mathrm{d}\boldsymbol{\phi}. \tag{5}$$

The expected conditional entropy, $\mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\left[\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}]\right]$, is the average entropy of the output $\mathbf{x}$ for a particular choice of inputs $\boldsymbol{\theta}$ and emulator parameters $\boldsymbol{\phi}$, under the posterior distribution of emulator parameters $p(\boldsymbol{\phi}|\mathcal{D})$. Again, we treat ensemble members $\boldsymbol{\phi}_m$ as if they were draws from $p(\boldsymbol{\phi}|\mathcal{D})$. Houlsby et al. refer to this rule as Bayesian Active Learning by Disagreement (BALD): we query parameters $\boldsymbol{\theta}$ where the posterior predictive is very uncertain about the output (entropy is high), but the emulator, conditioned on the value of its parameters $\boldsymbol{\phi}$, is on average quite certain about the model output (conditional entropy low on average).

For many distributions closed-form expressions of $\mathbb{H}\left[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}\right]$ are available, but this is in general not true for the entropy of the marginal predictive distribution $\hat{p}(\mathbf{x}|\boldsymbol{\theta},\mathcal{D})$. To overcome this problem, we derived an upper-bound approximation to the entropy term based on the law of total variance: if we characterize the marginal distribution only in terms of its (co)variance $\Sigma_{\mathcal{D}}(\boldsymbol{\theta})$, then $\mathbb{H}[\mathbf{x}|\boldsymbol{\theta},\mathcal{D}] \leq \frac{1}{2}\ln\left[(2\pi e)^N|(\Sigma_{\mathcal{D}}(\boldsymbol{\theta}))|\right]$. Using the law of total (co)variance, we get

$$\Sigma_{\mathcal{D}}(\boldsymbol{\theta}|\mathcal{D}) = \mathrm{Cov}[\mathbf{x}|\boldsymbol{\theta}] = \mathbb{E}_{\boldsymbol{\phi}|\mathcal{D}}\left[\mathrm{Cov}[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}]\right] + \mathrm{Cov}_{\boldsymbol{\phi}|\mathcal{D}}\left[\mathbb{E}[\mathbf{x}|\boldsymbol{\theta},\boldsymbol{\phi}]\right], \tag{6}$$

where all expectations can be approximated by samples drawn from $p(\boldsymbol{\phi}|\mathcal{D})$.

References

S Depeweg, J M Hernández-Lobato, F Doshi-Velez, and S Udluft. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *arXiv:1710.07283*, 2017.

Y Gal, R Islam, and Z Ghahramani. Deep bayesian active learning with image data. *arXiv:1703.02910*, 2017.

N Houlsby, F Huszar, Z Ghahramani, and M Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

Appendix D. Gaussian simulator example

D.1. Model

Data is generated independently according to $\mathbf{x}_i \sim \mathcal{N}(\cdot|f(\boldsymbol{\theta}), \boldsymbol{\Sigma})$, $i = 1 \dots n$, where $n = 10$, $f(\boldsymbol{\theta}) = (1.5\ \boldsymbol{\theta} + 0.5)^3/200$, $\boldsymbol{\Sigma}_{ii} = 0.1$, $\boldsymbol{\Sigma}_{ij} = 0$ for $i \neq j$, $\bar{\mathbf{x}}_o = \frac{1}{n}\sum_i^n \bar{\mathbf{x}}_o^{(i)} = \mathbf{2}$, and $\boldsymbol{\theta}$ is distributed uniformly in $[-8, 8]^p$ where p is the dimensionality of the problem.

This problem is inspired by the Gaussian example studied in Järvenpää et al. (2017), where f was chosen as $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$. We introduce a nonlinearity in f, since our method with uniform acquisitions would otherwise trivially generalize across the space – we observed that a neural network with the right amount of ReLu units can learn the linear mapping perfectly, independently of where the training samples are acquired.

D.2. Evaluation

We evaluate our method and BOLFI (Järvenpää et al., 2017) on this problem in 1D and 2D. In 1D, algorithms start with $t_0 = 10$ initial samples, in 2D with $t_0 = 25$, and make 100 acquisitions after each of which we evaluate how well the ground truth posterior is recovered.

As performance metric, we calculate total variation (TV) between $\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o)$ and $p(\boldsymbol{\theta}|\mathbf{x}_o)$, defined as

$$\frac{1}{2}\int \Big|\hat{p}(\boldsymbol{\theta}|\mathbf{x}_o) - p(\boldsymbol{\theta}|\mathbf{x}_o)\Big| \mathrm{d}\boldsymbol{\theta}.$$

D.3. Network architecture and training

Emulator networks model a normal distribution as output, so that the outputs of the network parametrise mean and covariance (Cholesky factor of the covariance matrix). Neural networks have one hidden layer consisting of 10 tanh units. We train an ensemble of $M = 50$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD, and a learning rate of 0.01.

D.4. BOLFI

BOLFI requires choice of a distance function: We use the the Mahalanobis distance

$$\Delta_{\boldsymbol{\theta}} = \big((\bar{\mathbf{x}} - \bar{\mathbf{x}}_o)^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{x}}_o)\big)^{1/2},$$

in line with the distance function used for the Gaussian example studied in Järvenpää et al. (2017). We use the implementation provided by the authors (Lintusaari et al., 2017).

References

M Järvenpää, M U Gutmann, A Pleska, Vehtari, A, and P Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *arXiv:1704.00520v2*, 2017.

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

J Lintusaari, H Vuollekoski, A Kangasrääsiö, K Skytén, M Järvenpää, M Gutmann, A Vehtari, J Corander, and S Kaski. Elfi: Engine for likelihood free inference. *arXiv:1708.00707*, 2017.

Appendix E. Image example

E.1. Model

Images are generated according to:

$$\begin{aligned} I_{xy} &\sim \mathrm{Bin}(\cdot|255, p_{xy}) \\ p_{xy} &= 0.9 - 0.8 \exp^{-0.5\left(r_{xy}/\sigma^2\right)^{\gamma}} \\ r_{xy} &= (x - x_{\mathrm{off}})^2 + (y - y_{\mathrm{off}})^2, \end{aligned}$$

where x and y are coordinates in the image, and $\mathrm{Bin}(\cdot|n, p)$ is the binomial distribution.

Model parameters are x_{off} and y_{off}, which respectively determine the horizontal and the vertical offset of the blob, γ, defining its contrast, and σ^2, determining the width width.

For our experiments, we use images of size 32×32 pixels. We choose uniform priors in the range $[-16, 16]$ for x_{off} and y_{off}, and a uniform prior in the range $[0.25, 5]$ for γ. We fix σ to 2.

E.2. Evaluation

We evaluate different acquisition methods by keeping track of the log-likelihood of a test set consisting of 100 parameters-image pairs over the course of acquisitions (starting from an initial sample of size $t_0 = 30$). In addition, we provide posterior predictive checks for an amortized emulator after acquiring $t = 1000$ samples.

E.3. Network architecture and training

Emulator networks model a binomial distribution as output. Neural networks have two hidden layers (200 units each) with ReLu activation functions. We train an ensemble of $M = 25$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD with a learning rate of 0.001.

References

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

Appendix F. Hodgkin-Huxley example

F.1. Model

The dynamic equations describing the evolution of the membrane potential and of the gating variables of the neuron are taken from Pospischil et al. (2008):

$$\begin{aligned} C_m\dot{V} &= -(I_{\text{leak}} + I_{\text{Na}} + I_{\text{K}} + I_{\text{M}} + I_{\text{ext}}) \\ &= g_{\text{leak}}(E_{\text{leak}} - V) + \bar{g}_{\text{Na}}m^3h(E_{\text{Na}} - V) + \\ &+ \bar{g}_{\text{K}}n^4(E_{\text{K}} - V) + \bar{g}_{\text{M}}p(E_{\text{K}} - V) + I_{\text{in}}(t), \end{aligned}$$

where C_m is membrane capacitance, V the membrane potential, I_c are ionic currents ($c = \{\text{Na}, \text{K}, \text{M}\}$) and $I_{\text{in}}(t)$ is an externally applied current which we can imagine as the sum of a static bias I_{bias} and a time-varying zero-mean noise signal $\varepsilon(t)$. I_{Na} and I_{K} shape the up- and down-stroke phases of the action potential (spike), I_{M} is responsible for spike-frequency adaptation, and I_{leak} is a leak current describing the passive properties of the cell membrane. Each current is in turn expressed as the product of a maximum conductance ($\bar{g}_c$) and the voltage difference between the membrane potential and the reversal potential for that current(E_c), possibly modulated by zero or more 'gating' variables (m, h, n, p).

Each $x \in \{m, h, n, p\}$ evolves according to first order kinetics in the form:

$$\dot{x} = \frac{1}{\tau_x(V)}\big(x_\infty(V) - x\big)$$

We provide a step current as input.

In our example application, free model parameters are g_{Na} and g_{K}. We model uniform priors over these parameters: $\bar{g}_{\text{Na}}$ is between 0.5 and 60 and $\bar{g}_{\text{K}}$ is between 0.5 and 10.

F.2. Evaluation

We evaluate the posterior obtained through the emulator after $t = 250$ acquisitions, starting from an initial sample size $t_0 = 30$. As posterior predictive check, we span a grid over the parameter space and compare simulator outputs to the posterior.

F.3. Network architecture and training

Emulator networks model a categorical distribution with $K = 6$ classes as output. Neural networks have two hidden layer (200 units each) with a ReLu activation functions. We train an ensemble of $M = 25$ networks using Adam (Kingma and Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) for SGD with a learning rate of 0.001.

References

D P Kingma and J Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980*, 2014.

M Pospischil, M Toledo-Rodriguez, C Monier, Z Piwkowska, T Bal, Y Frégnac, H Markram, and A Destexhe. Minimal hodgkin-huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99(4-5), 2008.

Appendix G. BOLFI convergence

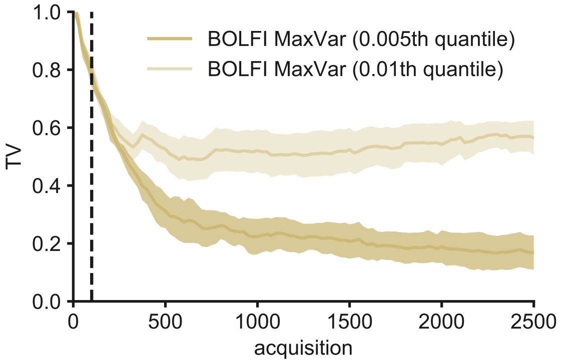

Figure 4: **Convergence of BOLFI *MaxVar*.** In the manuscript, we show performance up to 100 acquisitions (indicated by the dotted line). With additional acquisitions, BOLFI converges. The quality of the inferred posterior strongly depends on the value of the threshold hyperparameter used in BOLFI.

Appendix H. *MaxMI* acquisition for Hodgkin-Huxley model

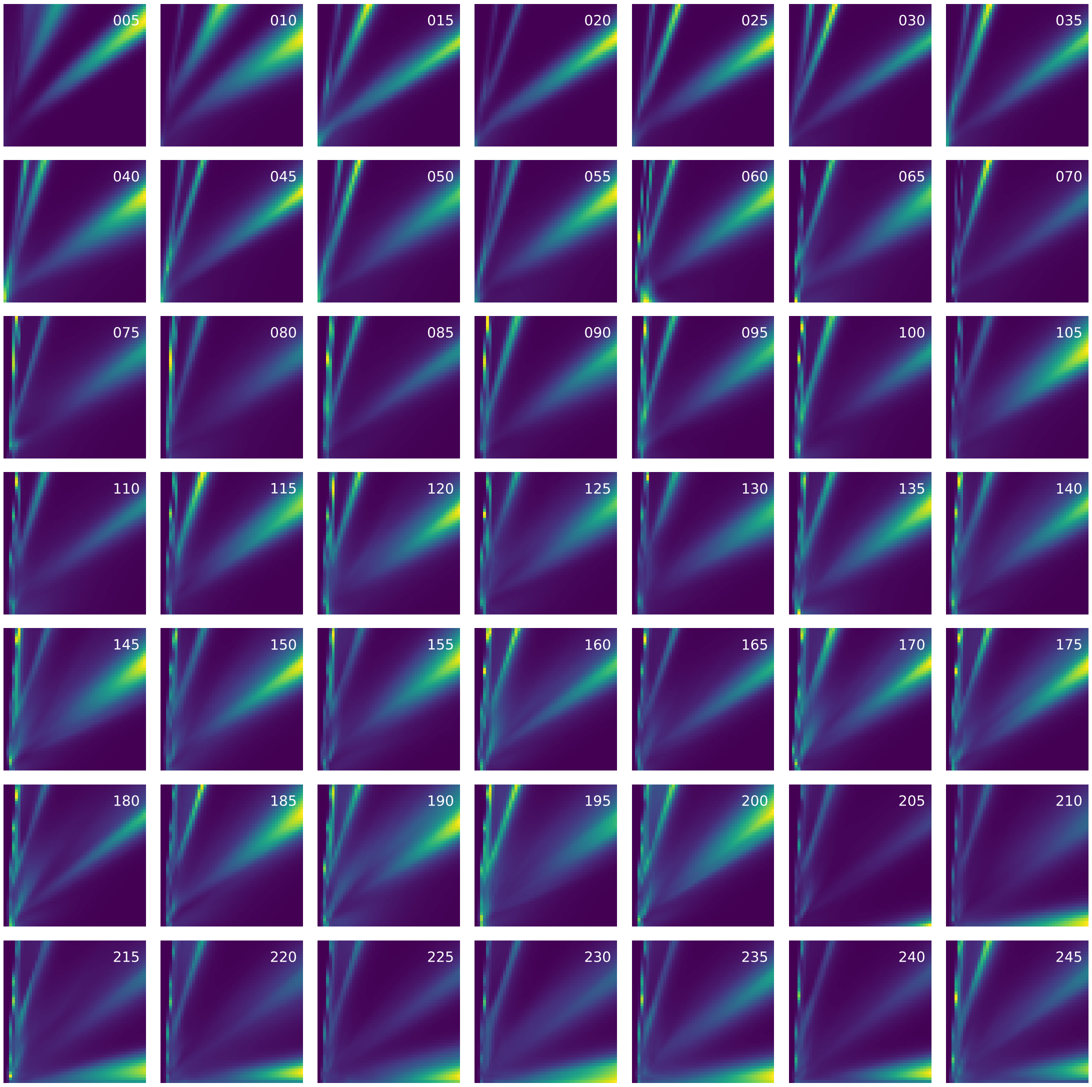

Figure 5: **Acquistion surface for *MaxMI* rule on Hodgkin-Huxley example.** Individual panels show the acquisition surface over $\boldsymbol{\theta}$ as additional samples have been acquired. The acquisition rule proposes datapoints at the decision boundaries of the posterior.