Machine Learning Surrogates for Highly Realistic Simulations

Patrick Cannon, Rory Greig, Gioia Boschi, Christoforos Anagnostopoulos
Improbable
London, UK
{patrickcannon, rory}@improbable.io

ABSTRACT

High-fidelity models of real-world systems such as infrastructure and civilian behaviour are the next frontier in improving the depth and sophistication of military simulations. Such complex systems are difficult to model with complete accuracy, and often involve unobserved free parameters. To make these simulations as realistic as possible, their parameters must be tuned by matching the simulation output to real-world data in a process known as 'calibration'. Typically, calibration is approached via brute force exploration of the parameter space in an attempt to identify the parameter settings that best reproduce patterns seen in data or expected by subject matter experts. However, with large numbers of parameters and a computationally expensive simulator, exhaustive approaches quickly become infeasible.

In structural engineering and elsewhere, this problem is often overcome using 'surrogate-based' optimisation, wherein a computationally efficient surrogate model is trained on sample input-output pairs from the simulation and is thereafter used for rapid parameter exploration. Application of surrogate methods enables fast exploration of parameters of interest, resulting in a simulator that produces more realistic results than are found through brute-force tuning. In addition, surrogates are amenable to training through 'active learning', where information gained from a single round of training is used to inform how the input space is explored in subsequent rounds. We illustrate the use of surrogates on an epidemiological agent-based model that has been highly influential in the UK response to the Covid-19 pandemic.

ABOUT THE AUTHORS

Dr. Patrick Cannon is a Research Scientist in the Complexity Research team at Improbable. He holds a PhD in mathematics from the University of Bristol and is chiefly interested in computational statistics and machine learning.

Rory Greig is a Research Scientist in the Complexity Research team at Improbable, with a background in software engineering. He specialises in calibration of agent-based models and holds a master's degree in electronic engineering.

Gioia Boschi is a PhD candidate in applied mathematics at King's College London. She is part of the Cross-disciplinary Approaches to Non-Equilibrium Systems (CANES) programme and currently works on opinion dynamics and collective memory. During her PhD she joined Improbable for an internship.

Dr. Christoforos Anagnostopoulos holds degrees in mathematics and theoretical computer science. Following a PhD in streaming data analysis, he was a lecturer in statistics at Imperial College London where he taught graphical modelling. After five years running a niche consulting startup in the area of cyber security, he joined Improbable as a Senior Principal Scientist. He has since returned to Imperial College London, where he teaches ethics of data science and AI.

Machine Learning Surrogates for Highly Realistic Simulations

Patrick Cannon, Rory Greig, Gioia Boschi, Christoforos Anagnostopoulos
Improbable
London, UK
{patrickcannon, rory}@improbable.io

INTRODUCTION

Sophisticated, large-scale simulations are increasingly used for decision making due to their expressiveness and explanatory power. Of particular recent popularity are *agent-based models* (ABMs), a broad class of generative models in which one specifies the behaviour of individual entities, or agents, and allows them to interact with each other and their environment. Though the individual behaviour may be simple, the simulation as a whole can exhibit complex emergent behaviour (Gatti et al., 2018). Such models appear in fields as diverse as military planning (T. M. Cioppa et al., 2004), ecology (Revilla, 2019), transportation (Wang et al., 2016) and economics (Baptista et al., 2016; Deissenberg et al., 2008).

As with any scientific model, it is desirable to be able to tune or *calibrate* such simulators by incorporating available data. Calibrating ABMs is however a highly non-trivial task. They are typically stochastic, with a complex, non-linear structure. A further challenge is the high computational cost of simulating ABMs, a consequence of their large number of agents interacting repeatedly over many timesteps or events, which can easily lead to run times for a single simulation of minutes or hours.

For some ABMs, it is feasible for a domain expert to provide certain parameter values in an ad-hoc fashion. Other parameters may be specific to the simulator or real-world situation considered, and so no precedent exists for their value. Even where domain expertise is available, principled calibration can provide complementary insight.

We present a method for calibrating ABMs which leverages regression strategies from machine learning in order to provide, within a fixed computational budget, better parameter estimates than are found through standard brute force approaches. The remainder of the paper is structured as follows. We first discuss the theoretical background to the calibration task, as well as some optimisation approaches that have been proposed. In the following section we describe our surrogate methodology and test it against the most common alternatives in computational experiments. Finally, we conclude and consider further work.

CALIBRATING SIMULATORS

In this section we describe a mathematical framework for simulator calibration and discuss some common approaches taken.

Framework and assumptions

Consider a model, or simulator, $\mathcal{M}$ with input parameters $\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_p\right)$ and output $\boldsymbol{x} = \mathcal{M}(\boldsymbol{\theta})$. For this exposition, $\boldsymbol{\theta}$ is a vector and $\boldsymbol{x}$ is one or more univariate time-series, however the framework presented here is quite general and the output type can easily be tailored to the model of interest. Parameters may include initial conditions of the simulator as well as values controlling its behaviour as it evolves. The simulator may be stochastic, in that for a fixed choice of parameters $\boldsymbol{\theta}$, the output generated by the model $\mathcal{M}(\boldsymbol{\theta})$ can vary and has some probability distribution. Where this fact is relevant, we will write $\boldsymbol{x} \sim \mathcal{M}(\boldsymbol{\theta})$ to indicate that the output $\boldsymbol{x}$ is a random variable drawn from the distribution implied by the simulator $\mathcal{M}(\boldsymbol{\theta})$.

It is worth noting that to refer to 'the output' of an ABM is vague and amounts to a modelling decision. At one extreme, the entire *trace* of the simulation could be reported; that is, the value taken by every variable at each time step. At the other, a single value from a single agent could be reported. Whatever choice is made regarding the output $\boldsymbol{x}$, it is conventional to report instead a carefully chosen set of summary statistics $s(\boldsymbol{x})$. This confers two benefits to the modeller. Firstly, it can offer significant dimensionality reduction, and secondly it allows the modeller to distil the essential features of the full output. As an example, when the output is a univariate time-series, the modeller may select a vector of summary statistics $s(\boldsymbol{x}) = \big(\text{mean}(\boldsymbol{x}), \text{variance}(\boldsymbol{x})\big)$. We take such decisions to be implicit in the definition of the model output hereafter and simply refer to the output $\boldsymbol{x}$.

To aid our explanation, we will use in this section and the next a running example of a simple epidemiological simulation, namely a Susceptible-Infected-Recovered (SIR) model (e.g. Hethcote, 2000). This is an example of a *compartmental* model, in which the interactions of a population are modelled not at the individual level, but by the proportion belonging at a given time to a particular category, e.g. the *infected* population I. Thus, it is not an agent-based model and is simpler than a typical ABM, with fewer parameters. Simulation experiments in the penultimate section will demonstrate calibration on two, more complex models. The SIR model is parameterised by $\boldsymbol{\theta} = (\beta, \gamma)$, the population contact and disease recovery rate respectively. It produces as output *epidemic curves*, in this case the number of individuals over time in each compartment (S, I or R). Figure 1 shows three examples of an infection (I) curve for differing values of $\boldsymbol{\theta}$.

In this work, calibration refers specifically to identifying the parameter values $\boldsymbol{\theta}$ for which the simulator $\mathcal{M}(\boldsymbol{\theta})$ produces output as close as possible to a time-series observation $\boldsymbol{x_o}$. In our running example, this amounts to determining which settings of $\boldsymbol{\theta} = (\beta, \gamma)$ lead the SIR model to produce epidemic curves as close as possible to observed, or desired, data curves such as the number of deaths or, as shown in Figure 1, active infections.

We now clarify some further assumptions that are held for the remainder of the paper. Firstly, we suppose it is possible to sample from the simulator $\mathcal{M}$ given a parameter vector $\boldsymbol{\theta}$, but that nothing is known about its internal workings. This precludes the use of any methods requiring gradients of the simulator output with respect to its input parameters. It also precludes the use of many traditional statistical methods, including maximum likelihood estimation (MLE) and Bayesian inference, since they rely on access to the model *likelihood*; the probability distribution of the output $\mathcal{M}(\boldsymbol{\theta})$ conditional on given input parameters $\boldsymbol{\theta}$. Nearly all realistically complex ABMs have intractable likelihoods. So-called *approximate inference* techniques in statistics show promise as generic methods for tackling intractable simulators. Families of techniques like approximate Bayesian computation (e.g. Beaumont, 2019), synthetic likelihood (Wood, 2010), and neural density estimation (e.g. Papamakarios, 2019) have proven to be successful in a range of challenging settings and will be explored in future work.

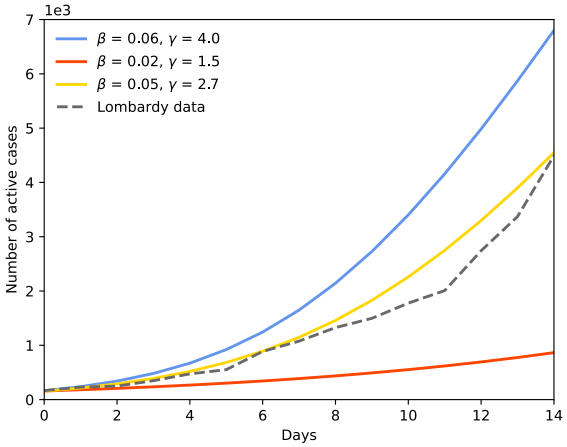

Figure 1. A comparison of SIR model infection curves with real infection data. The smooth lines are epidemic curves generated by the SIR model under three different parameter settings. Active cases of Covid-19 reported in Lombardy, Italy from the 20th of February 2020 are given by the dashed grey line.

In line with our interest in large-scale, complex ABMs we further assume that it is computationally expensive to perform a single evaluation of $\mathcal{M}$. More precisely, a single evaluation of the model is assumed to be orders of magnitude more computationally expensive than any of the ancillary operations required to generate parameter samples, return the minimum of a vector and so on, so that the cost of an entire optimisation algorithm is dominated by sampling from the model. Finally, we assume that a computational budget of n model samples is feasible and that the cost of a single model run is fixed.

In what follows, we consider two main approaches to calibration: direct black-box optimisation of the model and optimisation of a surrogate.

Optimisation approaches

It is possible to translate the calibration problem into an optimisation problem by describing the similarity of the model output to observed data by a *loss function* $f(\boldsymbol{\theta})$. An example of such a loss function is $f_{\boldsymbol{x_o}}(\boldsymbol{\theta}) \coloneqq L\big(\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{x}_o)\big)$ where $L(\boldsymbol{x}, \boldsymbol{x'}) \coloneqq ||\boldsymbol{x'} - \boldsymbol{x}||$ and $\boldsymbol{x_o}$ is observed data. With this characterisation, the calibration task can be cast as the standard optimisation problem

$$\text{argmin}_{\boldsymbol{\theta}}\, f_{\boldsymbol{x_o}}(\boldsymbol{\theta}) = \text{argmin}_{\boldsymbol{\theta}}\, L(\mathcal{M}(\boldsymbol{\theta}), \boldsymbol{x_o}). \quad (1)$$

The choice of loss function is an important modelling decision. Along with the choice of summary statistics $s(\boldsymbol{x})$, it should capture the modeller's belief about the most meaningful form of divergence from observed data. Having defined a relevant function $f(\boldsymbol{\theta})$ to minimise, we describe common strategies for doing so. Many popular optimisation techniques make use of gradient information, that is, quantities related to $\mathrm{d}f(\boldsymbol{\theta})/\mathrm{d}\boldsymbol{\theta}$. As gradient information is unavailable for most ABMs, we consider *black-box* optimisation techniques which aim to minimise a function f using only information gained from pointwise, possibly noisy, evaluations.

Perhaps the simplest method is grid search, in which the user picks a set of values for each parameter vector dimension, e.g. $\theta_i \in \{0{,}0.1, \dots, 1\}$ for $i = 1, \dots, p$, and the loss function is evaluated at every parameter combination. While the technique is simple and can be implemented in a parallel fashion, it quickly becomes computationally infeasible for large parameter dimension p as the number of combinations, and thus run time, is exponential in p.

An alternative to grid search is offered by sampling algorithms, the most well-known of which is Monte Carlo. In a standard Monte Carlo sampling method, the candidate parameter values are pseudorandom numbers drawn from a user-specified distribution, for example a uniform distribution over the parameter space. Such algorithms are traditionally used to estimate expectations, doing so with an error that diminishes as the number of samples increases at a rate independent of the parameter dimension p, but are nonetheless impracticably slow for many applications.

A competitor to standard Monte Carlo is Latin hypercube sampling (McKay et al., 1979), in which stratified sampling of the user-specified distribution is used to explore the parameter space more efficiently, avoiding collinear points. It has become a standard tool in statistics and related areas as a variance reduction technique in the stochastic estimation of integrals (Stein, 1987). Experiments in this work use a standard Latin hypercube design, though more elaborate schemes have been developed and are in some settings more efficient (e.g. Cioppa, 2001).

The sampling methods described in this section are sometimes described as 'brute force' approaches on account of their straightforward logic and substantial computational demands. Naturally, this shortcoming is compounded for large-scale, complex simulations like ABMs that are themselves computationally expensive to run. One might ask whether the increasing availability and affordability of cloud computing platforms goes some way to resolving this issue. We note that regardless of available computing power, it is always desirable to minimise the resources required to carry out a given computational task, for cost efficiency and environmental reasons among others.

SURROGATE-BASED CALIBRATION

In this section, we consider a family of methods in which samples are taken from the model not in order to optimise directly, but to train a *surrogate* for the original model.

In a typical supervised learning application, algorithms are trained on datasets of input-output pairs with the aim of achieving high predictive accuracy on previously unseen, unlabelled data. Synthetic training data can be constructed in a similar fashion for a simulator. Supposing a training dataset of size n is sought, parameter vector values $(\boldsymbol{\theta^i})_{i=1,\dots,n}$ can be obtained by, for example, any method described in the previous section. Each such parameter vector constitutes the *input* of a dataset pair. The *output* consists of the simulator $\mathcal{M}$ evaluated at this input. In this way, we can collect a training dataset of model parameter-output pairs $(\boldsymbol{\theta^i}, \boldsymbol{x^i})_{i=1,\dots,n}$, where $\boldsymbol{x^i} \sim \mathcal{M}(\boldsymbol{\theta^i})$ and we note '$\sim$' is replaced by

'=' for a deterministic simulator. This already captures the essence of the surrogate approach. However, since in this work we are interested particularly in loss minimisation, and in order to reduce the complexity of the output, we in fact take one further step. Given an input $\boldsymbol{\theta}$, and data $\boldsymbol{x_o}$, we sample $\boldsymbol{x} \sim \mathcal{M}(\boldsymbol{\theta})$, and take as output its loss with respect to the data $L(\boldsymbol{x}, \boldsymbol{x_o})$, giving finally a synthetic training dataset of parameter-loss pairs $\left(\boldsymbol{\theta^i}, L(\boldsymbol{x^i}, \boldsymbol{x_o})\right)_{i=1,\dots,n}$. This reduces the dimensionality of the training sample output from one or more univariate time-series to a scalar.

Our aim is to use a machine learning algorithm to generalise patterns seen in this limited budget of model evaluations, to infer candidate parameter values that are likely to have lower loss, even if the model has not yet been evaluated on them. In machine learning, this task is described as *regression* and the trained machine learning algorithm as a *regressor*. We term this trained regressor a *surrogate* model, as after training on synthetic data from the model it can act as a substitute for the true model. An existing example of a similar approach is shown in Lamperti et al. (2018), where an *XGBoost* model (Chen & Guestrin, 2016) is used as a surrogate.

There are several advantages to using a machine learning surrogate in place of the true model. Our principal motivation is that, in realistic scenarios, querying the surrogate is orders of magnitude faster than sampling from the original model. This facilitates better use of calibration methods than would be possible with the original simulator, albeit on an approximate model. The surrogate method takes advantage of this, after training on a small number of sample pairs $\left(\boldsymbol{\theta^i}, L(\boldsymbol{x^i}, \boldsymbol{x_o})\right)_{i=1,\dots,n}$, by querying the surrogate (to predict the loss) at many more parameter vector values $(\boldsymbol{\theta^i})_{i=1,\dots,n'}$ where n' can be orders of magnitude larger than n. Provided the surrogate has learned a reasonable approximation to the map $\boldsymbol{\theta} \mapsto L(\mathcal{M}(\boldsymbol{\theta}), \boldsymbol{x_o})$, when queried on the set $(\boldsymbol{\theta^i})_{i=n+1,\dots,n'}$, it is likely to be able to return a parameter vector with a lower loss than was present in the original training samples.

As an illustration of the surrogate method, we show in Figure 2 the true loss surface for the SIR model example introduced above. Superimposed are dots representing candidate parameter values generated using a simple grid search. In this case, a random forest regressor (Tin Kam Ho, 1995) was able to learn the parameter-loss mapping with exactly the same samples to an accuracy sufficient to outperform grid search. For higher dimensional problems the loss surface is often considerably more complex than this example, perhaps lacking smoothness and featuring multiple local minima, making the greater number of samples afforded through use of a surrogate a considerable advantage.

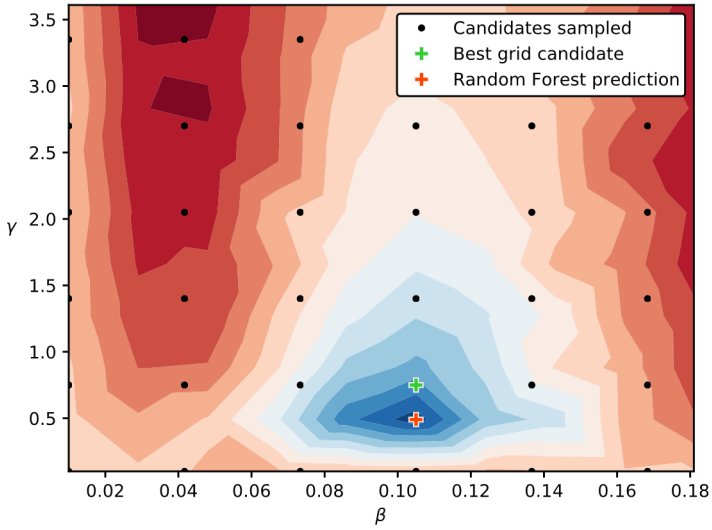

Figure 2. A random forest SIR surrogate. Here, a random forest regressor successfully interpolates between model evaluations of an SIR model, identifying a better solution (red cross) than grid search on a candidate set of 20 points was able to produce (green cross). Here, the contour plots indicate the real loss surface evaluated offline, with blue indicating regions of lower loss (i.e. matching the data better) and red indicating regions of higher loss.

In this work, we establish the pragmatic advantage conferred by the use of surrogates to achieve comparable performance to brute force search at a fraction of the computational cost. In the concluding section we discuss some

further advantages of surrogates, such as their use as proxies for the model in statistical estimation procedures, or amortisation, which we do not pursue here.

Surrogate models

In the experiments conducted below, several machine learning algorithms are used as surrogates to learn the loss function of the underlying model and data. The relative calibration performance of the following surrogates is investigated:

- **Gradient boosted trees** are an ensemble technique consisting of multiple decision trees that sequentially correct the error or *residuals* of the preceding trees. Here, we use the XGBoost library (Chen & Guestrin, 2016) to implement gradient boosted trees. We used 30 trees for our experiments.

- **Neural networks** are a family of highly flexible function approximators. The specific type of neural network model used in this paper is a multilayer perceptron (MLP) regressor, which is a simple neural network with a configurable number of layers of neurons. Multilayer neural networks are usually trained with variants of stochastic, gradient-based optimisers (e.g. Goodfellow et al., 2016); in this case *Adam* (Kingma & Ba, 2014) was used. For the general stochastic epidemic model, described below, we used a neural network with two hidden layers of size four and a learning rate of 0.002. For the Covid-19 ABM we used two hidden layers of size eight and the same learning rate.

- k**-Nearest Neighbours** (k-NN) is a regression technique in which the output predicted for a given input is based on the average of the training points closest to that input. The constant k determines the number of 'close' points used for this average. The implementation of k-NN used in this paper automatically chooses between three underlying algorithms to calculate the nearest neighbours: *Brute Force*, *K-D Tree* (Bentley, 1975) and *Ball Tree* (Omohundro, 1989). This automatic selection is based on the number of samples and dimensionality of the data. For our experiments we used $k = 5$.

SIMULATION STUDIES

In this section, experiments are performed to compare brute force search procedures with machine learning surrogates.

The first experiment concerns the general stochastic epidemic (GSE) model (Bailey, 1975) simulated following Sellke (1983). In the second experiment, surrogate calibration methods are tested on the *OpenABM* Covid-19 agent-based model (Fraser, 2020). This is a large-scale ABM capable of modelling interactions between individuals at the scale of a country. Additionally, it is capable of modelling the effect of interventions such as a lockdown. We test this model using *pseudotruth* simulations, explained in more detail below.

For both models the following factors were varied in order to understand the effects of each on calibration performance:

- *Family of technique* (direct optimisation or surrogate optimisation). The surrogate, once trained, is evaluated according to a brute force search algorithm, which itself has a fixed budget. All surrogates (XGBoost, neural net, k-NN) use $100{,}000$ surrogate evaluations.

- *Sampling method* (grid, uniform, Latin-hypercube). The choice of sampling method to obtain points in the parameter space. For brute-force techniques, these constitute the parameter vector values at which the model will be evaluated. For surrogates, they are coupled with the model evaluations at the same parameter vector values, which together defines the synthetic training set.

For the GSE model we used a model sample budget of $n = 10$ while for the more complex Covid-19 ABM we used a model sample budget of $n = 64$, the motivation for which we now explain. It is clear that with a sufficiently large number of samples brute force will likely outperform surrogates. However, the use of surrogates is intended to reduce the number of samples required to achieve a certain performance. To show this effect, we have selected a number of

samples that is small relative to the number of parameters, and scales (as with brute force) in the number of free parameters. That is, our choice of sample size was dictated by the number of parameters to be calibrated for each model (two for GSE and four for OpenABM) and not by the computational effort involved in running either model.

The distance metric used for all results in this paper was the Euclidean distance taken on the difference of the output time series, which performed reliably across a series of tests. The time series data used for the OpenABM model was cumulative deaths; taking the difference of this is equivalent to considering new deaths each day. Given that the GSE model does not describe the evolution of deaths over time, it has instead been calibrated using both the infection and recovery time series. In this case the loss function adopted was the average of the individual loss functions over these two time-series.

Results, such as those in Figure 4, are produced within the experimental framework by running multiple samples of each configuration. Samples are taken while varying the random seed provided to the model and sampling methods, which is necessary since the models are stochastic in nature.

Pseudotruth calibration

Pseudotruth calibration is a type of experiment used to measure the performance of a calibration method. First, we generate data from the model by running it with fixed parameter vector value (the choice of which is arbitrary). This model output is then taken as the ground truth, and we attempt to recover the parameter vector that was used to generate it. Success in this task provides some evidence of efficacy prior to calibration on real data, where the 'true' parameters are unknown. The advantage of this technique is that there is a clear measure of success, something that is much harder to quantify in the case of real data. The validation score used to measure accuracy in this experiment is the root mean squared error (RMSE) between the true parameters and best parameters found by the calibration method. It is important to emphasise that this RMSE score is distinct from the loss, which is calculated by comparing outputs, not inputs.

Experiments

GSE model

The GSE model is a stochastic SIR model. It is not an ABM and possesses a tractable likelihood function, from which we can derive the MLE of the model's parameters (Kypraios, 2007). This provides an opportunity to compare the parameters obtained through MLE, to those obtained through the calibration techniques, using the same real data. The validation score for this experiment was the RMSE between the parameter values found via MLE and the parameter values found by each surrogate technique.

As in the SIR model, the GSE model has parameters $\boldsymbol{\theta} = (\beta, \gamma)$. In order to set a reasonable upper bound for the *basic reproduction number* R_0 (namely $R_0 < 20$), the model was reparameterised to take $R_0 = \beta/\gamma$ instead of γ as a parameter. In the experiments, surrogate methods were calibrated using the *infection* and *removed* time-series (where 'removed' comprises recovered individuals and deaths), with no summary statistics applied. In addition, in order to estimate the parameters of this model with MLE, we also require the *susceptible* time-series. Given that the population is in reality not well-mixed, violating an assumption of the GSE model, we will assume that only 10% of the total population is initially susceptible. The susceptible time-series is then calculated by subtracting the infected and removed cases from this susceptible population.

The data is from the Covid-19 epidemic in Lombardy in March 2020. Given the lockdown measures implemented in this region, the GSE model is only appropriate for modelling the early phase of the epidemic, since this model does not support modelling the effects of policy interventions such as a lockdown.

Figure 3 shows time-series data from the real epidemic alongside data generated from the model, which was run using parameters calculated by MLE from the data ($\boldsymbol{\theta} = (\beta, R_0) = (0.28, 5.6)$). The time-series generated from the model with the MLE parameters seems to capture the behaviour of the true data for both infected and removed curves. Figure 4 compares the calibration performance of optimisation using surrogate methods and direct optimisation via brute force search on the GSE model. Note that some box plots with low variance appear only as solid black lines.

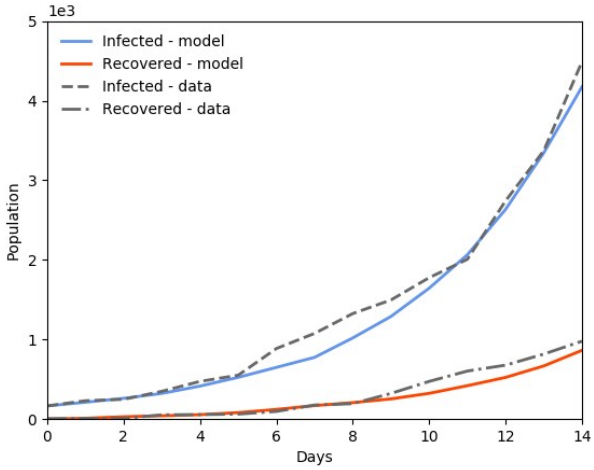

Figure 3. Time-series output for the GSE model. Active (resp. recovered) cases of Covid-19 reported in Lombardy, Italy from the 20th of February 2020 are given by the dashed (resp. dot-dash) grey line.

All surrogate methods trained on grid search samples outperform brute force grid search. Both Latin hypercube and uniform sampling, with direct brute force optimisation, offer an improvement over grid search. Only neural networks consistently outperform the brute force search baseline for all sampling methods.

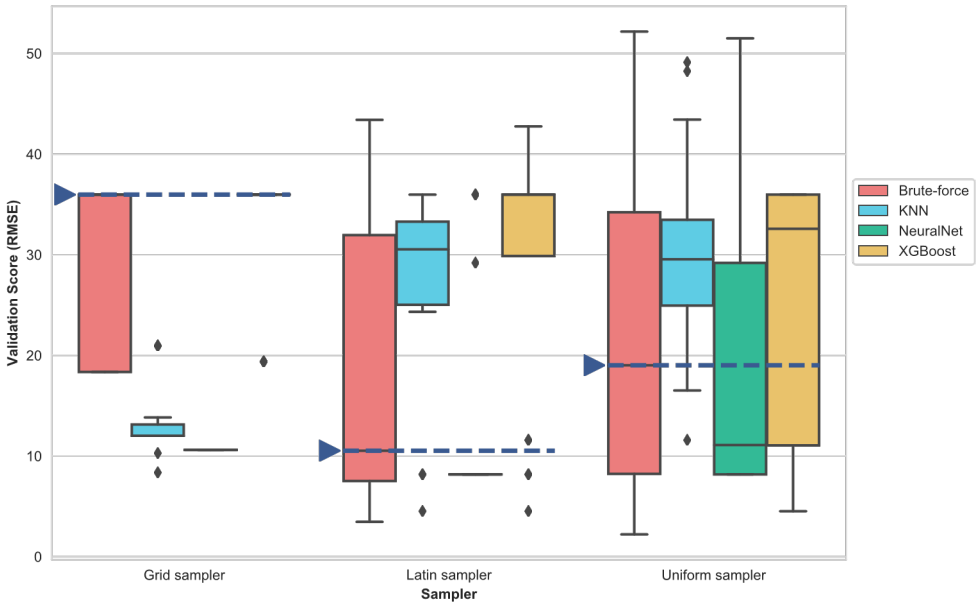

Figure 4. Calibration performance for the GSE model. A validation score is shown on the y-axis (the RMSE between the maximum likelihood estimate and surrogate-calibrated parameters) for the different methods; lower is better. On the x-axis are three groups of four boxplots, grouped by sampling method. The boxplot colour indicates the calibration method used (either direct brute force or with a specific surrogate method). Within each sampling method, the dotted blue line indicates the median performance of brute force search for that sampling method. If the median performance for the surrogate (indicated by a solid black line) is lower than the dotted blue line, then the surrogate has outperformed brute force search.

There are some limitations to this experimental setup which should be held in mind when drawing conclusions from these results. The first is that this simple model has only two parameters, so even a relatively small number of model evaluations can give fairly good performance with the brute force techniques such as grid search. We expect the surrogate methods to definitively outperform brute force techniques in higher dimensional calibration problems, particularly where there are rough loss surfaces. In higher dimensions it is expected that the surrogate techniques would require fewer samples to produce good calibration results than brute force techniques, which do not scale well to higher dimensions.

Covid-19 agent-based model

Figure 5 shows a pseudotruth experiment to calibrate four parameters of the OpenABM model: *infectious rate*, *mean infectious period*, *asymptomatic infectious factor* and *mild infectious factor*. The time-series used for pseudotruth calibration was cumulative deaths. We used a simulated population of 100,000 individuals for the experiments. All other parameters not being calibrated were set to their defaults found in Fraser (2020).

Brute force search techniques perform reasonably well as a baseline. Of the surrogate methods, the neural network seems to perform the best, outperforming the baselines for all sampling methods. The best performing combination overall is uniform sampling with the neural network surrogate. The *k*-NN surrogate also performs well, outperforming the brute force search baseline for the grid sampler and uniform sampler, and similar performance for the Latin hypercube sampler. All combinations of sampling methods and surrogates seem to outperform direct grid search.

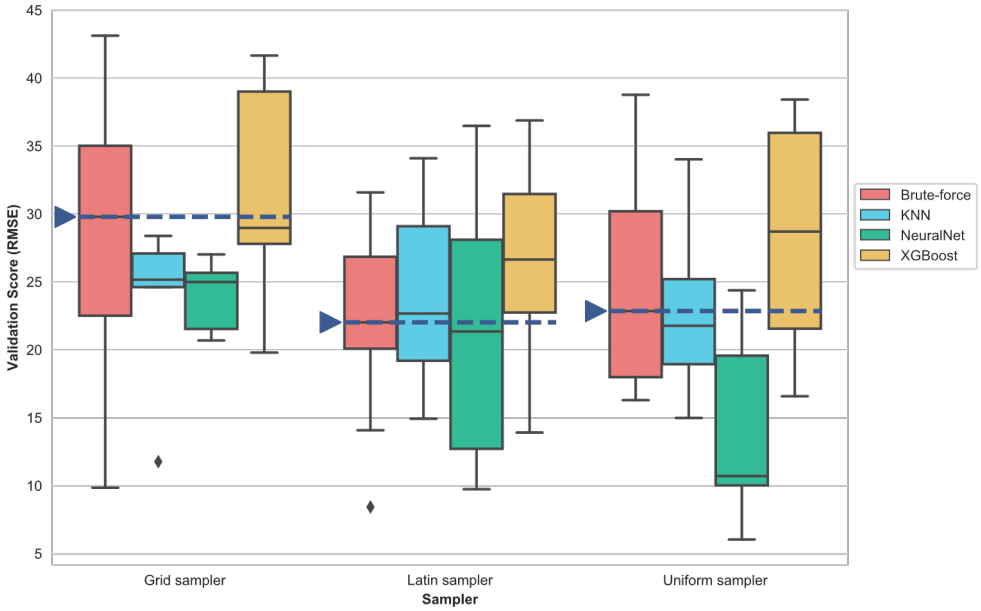

Figure 5. Calibration performance for the Covid-19 OpenABM model. The y-axis shows a validation score for each method, namely the RMSE between all four parameters and the true parameter values used to generate the pseudotruth data; lower is better.

An alternative training method for the surrogate is now briefly discussed. In Lamperti et al. (2018), an XGBoost model is trained using a so-called *active* learning approach (e.g. Settles, 2010), which uses successive rounds of training to produce the final surrogate. Here we elaborate on an alternative approach, using an ϵ-greedy method, but leave detailed computational experiments to future work.

We propose the following active learning algorithm. First, as before, an initial number of parameter-loss samples are used to train the surrogate. This surrogate is then used to judge the most profitable area of the parameter space from which to take further samples in subsequent rounds of training. In each round, new samples are taken and added to the total set of training samples, and the surrogate is retrained.

In particular, a proportion ϵ of the new samples are taken from the 'neighbourhood' of the best prediction of minimum loss from the surrogate trained in the previous round, and the remainder are taken, for example, from a uniform distribution across the whole parameter space. The proportion of global samples can be gradually reduced with each successive round, as the algorithm narrows in on the area of the best prediction. This can be accomplished by using a proportion ϵ_n in round n, where $\epsilon_n = 1 - 1/n$. In this way, one could achieve greater efficiency by balancing the *exploration-exploitation* trade-off. Exploration of the full parameter space is performed by the global samples, of which there is a greater proportion in the earlier rounds. In later rounds, the algorithm exploits more often the best current predicted minimum of the surrogate in order to gain more information in a neighbourhood of that minimum.

CONCLUSIONS AND FURTHER WORK

Several combinations of calibration techniques have been applied to an ABM of the current Covid-19 epidemic, as well as a GSE model which was calibrated using real data from the Covid-19 epidemic in Lombardy, Italy.

We have demonstrated tentative evidence from a pseudotruth calibration experiment that certain machine learning surrogates can outperform brute force techniques and deliver better calibration performance, even on complex ABMs with nonlinear behaviour. In particular the neural network surrogate performed well on both calibration problems, clearly outperforming brute force search. There are large differences in the performance of various machine learning surrogate methods, showing that the choice of surrogate technique is critical. Further work must be done to understand the optimal settings for these surrogates, for example through automated hyperparameter optimisation schemes.

The difference in performance of the various combinations of techniques shows that it is important to systematically compare different calibration methods in order to gain the full benefits of calibrating large-scale agent-based models.

The number of parameters calibrated in these experiments was relatively low (two for the SIR and GSE models, and four for the ABM), resulting in an optimisation task that is feasibly solved using brute force techniques. We have shown that even in this setting, surrogates show promising benefits over direct brute force methods. However, with higher dimensional parameter spaces it is expected that the surrogates will have a greater advantage, since brute force techniques scale poorly. Further work should focus on comparing performance of these methods on higher dimensional parameter spaces.

There are additional benefits to the use of machine learning surrogates which can be explored in future work.

Firstly, some machine learning algorithms are equipped with methods for estimating the model likelihood. Following training of the surrogate, this could be used directly to estimate parameters using MLE or other standard statistical approaches, without needing to query the surrogate on further points in the parameter space. Secondly, surrogate algorithms can be trained on parameter-loss pairs $\big(\boldsymbol{\theta}, L(\boldsymbol{x}, \boldsymbol{x_o})\big)$, as we have done in this work, but also on parameter-output pairs $\big(\boldsymbol{\theta}, \mathcal{M}(\boldsymbol{\theta})\big)$. In the second case, we can take advantage of *amortisation*, whereby the training procedure can be carried out once, in advance of the arrival of the real data (notice that no real data appears in the expression $\big(\boldsymbol{\theta}, \mathcal{M}(\boldsymbol{\theta})\big)$, and no further training must take place as new observations arrive. To be precise, once the surrogate is trained on a set of parameter-output pairs, we can easily use it to predict $\big(\boldsymbol{\theta}, L(\boldsymbol{x}, \boldsymbol{x_o})\big)$ for a new observation $\boldsymbol{x_o}$ by first evaluating the surrogate at $\boldsymbol{\theta}$ to give the predicted output $\boldsymbol{x}$ for this choice of input, then subsequently evaluating the loss $L(\boldsymbol{x}, \boldsymbol{x_o})$ of this predicted output. Finally, so-called *feature importance* techniques (see, e.g., Breiman, 2001) are available for some regressors, giving a quantitative understanding of the degree to which each input (parameter in our case) affects the output of the surrogate. This has obvious connections to model sensitivity analysis.

REFERENCES

Bailey, N. T. (1975). *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE.

Baptista, R., Hinterschweiger, M., Low, K., & Uluc, A. (2016). Macroprudential Policy in an Agent-Based Model of the UK Housing Market. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2850414

Beaumont, M. A. (2019). Approximate Bayesian Computation. *Annual Review of Statistics and Its Application*.

https://doi.org/10.1146/annurev-statistics-030718-105212
Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM*, *18*(9), 509–517. https://doi.org/10.1145/361002.361007
Breiman, L. (2001). Random Forests. *Machine Learning*. https://doi.org/10.1023/A:1010933404324
Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785
Cioppa, T. (2001). *Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models*.
Cioppa, T. M., Lucas, T. W., & Sanchez, S. M. (2004). Military applications of agent-based simulations. *Proceedings of the 2004 Winter Simulation Conference, 2004.*, *1*, 180.
Deissenberg, C., Van Der Hoog, S., & Dawid, H. (2008). EURACE: A massively parallel agent-based model of the European economy. *Applied Mathematics and Computation*, *204*(2), 541–552.
Fraser, C. (2020). OpenABM-Covid19: Agent-based model for modelling the Covid-19. In *GitHub repository*. GitHub. https://github.com/BDI-pathogens/OpenABM-Covid19
Gatti, D. D., Fagiolo, G., Gallegati, M., Richiardi, M., & Russo, A. (Eds.). (2018). *Agent-Based Models in Economics: A Toolkit*. Cambridge University Press.
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, *42*(4), 599–653.
Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.
Kypraios, T. (2007). *Efficient Bayesian Inference for Partially Observed Stochastic Epidemics and A New Class of Semi-Parametric Time Series Models* [PhD Thesis].
Lamperti, F., Roventini, A., & Sani, A. (2018). Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*. https://doi.org/10.1016/j.jedc.2018.03.011
McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239–245.
Omohundro, S. M. (1989). *Five balltree construction algorithms*. International Computer Science Institute Berkeley.
Papamakarios, G. (2019). *Neural Density Estimation and Likelihood-free Inference*.
Revilla, E. (2019). Individual and Agent-based Models in Population Ecology and Conservation Biology. *Population Ecology in Practice*.
Sellke, T. (1983). On the asymptotic distribution of the size of a stochastic epidemic. *Journal of Applied Probability*, *20*(2), 390–394.
Settles, B. (2010). Active learning literature survey. *Technical Report 55-66, University of Wisconsin, Madison*.
Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, *29*(2), 143–151.
Tin Kam Ho. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282 vol.1.
Wang, H., Mostafizi, A., Cramer, L. A., Cox, D., & Park, H. (2016). An agent-based model of a multimodal near-field tsunami evacuation: Decision-making and life safety. *Transportation Research Part C: Emerging Technologies*. https://doi.org/10.1016/j.trc.2015.11.010
Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*. https://doi.org/10.1038/nature09319