INSTITUTE
OF ECONOMICS

Scuola Superiore
Sant'Anna

LEM | Laboratory of Economics and Management

Institute of Economics
Scuola Superiore Sant'Anna

Piazza Martiri della Libertà, 33 - 56127 Pisa, Italy
ph. +39 050 88.33.43
institute.economics@sssup.it

LEM
WORKING PAPER SERIES

Agent-Based Model Calibration using Machine Learning Surrogates

Francesco Lamperti °*
Andrea Roventini °§
Amir Sani ¶°

° Institute of Economics, Scuola Superiore Sant'Anna, Pisa, Italy
* FEEM, Milan, Italy
§ OFCE-Sciences Po, Nice, France
¶ CFM, Imperial College LOndon, UK

2017/11 **October 2017**
ISSN(ONLINE) 2284-0400

Agent-Based Model Calibration using Machine Learning Surrogates

Francesco Lamperti[*], Andrea Roventini[†] and Amir Sani[‡]

October 3, 2017

Abstract

Efficiently calibrating agent-based models (ABMs) to real data is an open challenge. This paper explicitly tackles parameter space exploration and calibration of ABMs by combining machine-learning and intelligent iterative sampling. The proposed approach "learns" a fast surrogate meta-model using a limited number of ABM evaluations and approximates the nonlinear relationship between ABM inputs (initial conditions and parameters) and outputs. Performance is evaluated on the Brock and Hommes (1998) asset pricing model and the "Islands" endogenous growth model (Fagiolo and Dosi, 2003). Results demonstrate that machine learning surrogates obtained using the proposed iterative learning procedure provide a quite accurate proxy of the true model and dramatically reduce the computation time necessary for large scale parameter space exploration and calibration.

Keywords: agent based model; calibration; machine learning; surrogate; meta-model.

JEL codes: C15, C52, C63.

[*]Corresponding author. Institute of Economics, Scuola Superiore Sant'Anna, Piazza Martiri della Libertà 33, 56127 Pisa (Italy). Email: f.lamperti@santannapisa.it.

[†]Institute of Economics, Scuola Superiore Sant'Anna (Pisa) and OFCE - Sciences Po (Nice). Email: a.roventini@santannapisa.it.

[‡]CFM-Imperial Institute of Quantitative Finance, Imperial College London and Institute of Economics, Scuola Superiore Sant'Anna (Pisa). Email: reachme@amirsani.com.

1 Introduction

This work proposes a novel approach to model calibration and parameter space exploration in agent-based models (ABM). It combines supervised machine learning and intelligent sampling in the design of a *surrogate* meta-model, which constitutes a computationally cheap approximation of the real model.[1] Our surrogate can then be employed to explore the parameter space of the model at almost zero computational costs.

ABMs deal with the study of socio-ecological systems that can be properly conceptualized through a set of micro and macro relationships. One problem with this framework is that the relevant statistical properties are *a priori* unknown, even to the modeler. Such properties emerge from the repeated interactions among ecologies of heterogeneous, boundedly rational and adaptive agents.[2] This results in dynamic properties that cannot be studied analytically, causal mechanisms that are not always possible to identify and emergent relationships that cannot be deduced by simple aggregation of micro-level interactions (Anderson et al., 1972, Tesfatsion and Judd, 2006, Grazzini, 2012, Gallegati and Kirman, 2012). This raises the issue of finding appropriate tools to investigate the emergent behavior of the model with respect to different parameter settings, random seeds, and initial conditions (see also Lee et al., 2015).

The primary challenge in exploring the parameter space and calibrating ABMs is the escalation in the number of parameters resulting from increasingly realistic ABM dynamics. For example, recent macroeconomic models use dozens of parameters to capture the complexity of micro-founded, multi-sector and multi-country phenomena (see Fagiolo and Roventini, 2017, for a recent survey). Existing tools for direct estimation and global sensitivity analysis (often advocated as a natural approach to ABM exploration, cf. Moss, 2008; Thiele et al., 2014; ten Broeke et al., 2016) are computationally prohibitive, requiring time and computational resources that are not often available to researchers or practitioners. This increase in the parameter set results in what is referred to as the "curse of dimensionality", i.e. the convergence of any estimator to the true value of a smooth function defined on a high dimensional parameter space is very slow (Weeks, 1995; De Marchi, 2005). There are potentially an exponential number of local critical points in the parameter space that can be mistaken for global maxima or minima.

Traditionally, three computationally expensive steps are involved in ABM calibration; running the model, measuring calibration quality and locating parameters of interest (more on validation of ABMs in Fagiolo et al., 2017). As remarked in Grazzini et al. (2017), such steps account for more than half of the time required to estimate ABMs, even for extremely simple models. Appropriate tools need then to be designed to quickly search for "meaningful" parameters and initial conditions. One approach is to replace the computationally expensive ABM with a cheaper proxy. This is the aim of meta-models or surrogates, which approximate the relationship between ABMs' inputs and outputs (see Lee et al., 2015; Fagiolo et al., 2017) in order to quickly explore the parameter space. Surrogate

[1] Supervised learning is the machine learning task of inferring a function from training data, which consists in a set of input-output pairs. A supervised learning algorithm analyses the training data and produces an inferred function, which is called a classifier (if the output is discrete) or a regression function (if the output is continuous). Such a function is then used to make prediction over data-points outside the training sample. Intelligent sampling refers here to the task of selecting data-points, used to learn the surrogate, in a way they covey the maximal amount of information.

[2] In the last two decades a variety of ABM have been applied to study many different issues across a broad spectrum of disciplines beyond economics and including ecology (Grimm and Railsback, 2013), health care (Effken et al., 2012), sociology (Macy and Willer, 2002), geography (Brown et al., 2005), bioterrorism (Carley et al., 2006), medical research (An and Wilensky, 2009), military tactics (Ilachinski, 1997) and many others. See also Squazzoni (2010) for a discussion on the impact of ABM in social sciences, and Fagiolo and Roventini (2012, 2017) for an assessment of macroeconomic policies in agent-based models.

models are traditionally employed as fast approximations of complex phenomena that are expensive to evaluate in real life or in simulation (see Booker et al., 1999), and are regularly leveraged to locate promising parameter combinations avoiding costly computations. Accordingly, if the approximation error is small, the surrogate can be interpreted as a reasonably good replacement for the original ABM during parameter space exploration, calibration and sensitivity analysis.[3]

Recently, kriging (Rasmussen and Williams, 2006; Conti and O'Hagan, 2010) has been introduced as a surrogate modeling approach to facilitate parameter space exploration and sensitivity analyses of ABMs (Salle and Yildizoglu, 2014; Dosi et al., 2016, 2017c,b; Bargigli et al., 2016). However, when the model's response surface is completely unknown and possibly contains non-smooth regions, as it is typically the case in ABMs, kriging requires a large number of evaluations and extensive exploratory data analysis that increase with the size of the parameter space (more on that in Section 2). Such constraints hold also for state-of-the-art extensions (see Wilson et al., 2015; Herlands et al., 2015) and it forces modelers of large scale ABM to arbitrarily fix a subset of parameters whenever the parameter space is too large (see e.g. Barde and van der Hoog, 2017).

What is needed is an efficient, "hands-off" approach to explore the complex parameter space of agent-based models that practically accounts for the limited computational resources of the user. Our approach explores the ABM parameter space using a non-parametric machine learning surrogate and iterative sampling algorithm that intelligently searches the response surface with few limiting conditions. In particular, no parametric assumptions or knowledge of the topology governing the spatial distribution of the data is required.

In a nutshell, the procedure begins by first drawing a relatively large "pool" of parameter combinations using any standard sampling routine, where each combination contains a value for each initial condition. This pool acts as a proxy for the full parameter space. Next, a (very small) random subset of combinations are drawn without replacement from the pool to initialize the learning procedure (again using any standard sampling routine). The ABM is then evaluated for each of these initial combinations and its outputs receive a "label". Those outputs satisfying a user-defined calibration criterion are assigned to a "positive" category (label 1), otherwise to a "negative" one (label 0). A surrogate is then learned over the combinations using the selected surrogate algorithm.[4] The first surrogate is used to predict the probability that unlabeled combinations in the pool belong to the "positive" category. This concludes the first round. In the second and subsequent rounds, a very small subset of the pool is drawn according to the predicted positive probability. These selections are evaluated in the ABM to learn their true labels and aggregated to the set of all other combinations that have been sampled during the previous rounds. This continues over multiple rounds until the user-defined number of evaluations (the so called "budget") is reached or a predefined level of performance is achieved.

As illustrative examples, we apply our procedure to two well known ABMs: the asset pricing model proposed in Brock and Hommes (1998) and the endogenous growth model developed in Fagiolo and Dosi (2003). Despite their relative simplicity, the two models might exhibit multiple equilibria, allow different behavioural attitudes and account for a wide range of dynamics, which crucially depends on their parameters. We find that our machine-learning surrogate is able to efficiently filter out

[3]Note that surrogates can be used for sensitivity analysis if their approximation errors are very small. This requires many more evaluations than those used in the examples of this paper.

[4]In the paper, we choose to use a non-parametric machine learning algorithm, the extremely boosted gradient trees (XGBoost, see Chen and Guestrin, 2016) as our surrogate. However, the user can choose different surrogates such as simple logistic regression. More on that in Section 3.

combinations of parameters conveying the output of interest, assess the relative importance of models' parameters and provide an accurate approximation of the underlying ABM in a negligible amount of time. The advantages in terms of computation cost, hands-free parameter selection and ability to deal with non-linear characteristics of the ABM parameter space of our approach paves the way towards an efficient and user-friendly procedure to parameter space exploration and calibration of agent-based models.

The rest of the paper proceeds as follows. Section 2 reviews literature on ABM calibration validation, making the case for surrogate modeling. Section 3 presents our surrogate modeling methodology. Sections 4 and 5 report the results of its application to the asset pricing model proposed in Brock and Hommes (1998) and the growth model developed in Fagiolo and Dosi (2003) respectively. Finally, Section 6 concludes.

2 Calibration and validation of agent-based models: the case for surrogate modelling

As stated in Fagiolo et al. (2007) and Fagiolo and Roventini (2012, 2017), the extreme flexibility of ABMs concerning various forms of individual behaviour, interaction patterns and institutional arrangements has allowed researchers to explore the positive and normative consequences of departing from the often over-simplifying assumptions characterizing most mainstream analytical models. Recent years have witnessed a trend in macro and financial modeling towards more detailed and richer models, targeting a higher number of stylized facts, and claiming a strong empirical content.[5]

A common theme informing both theoretical analysis and methodological research concerns the relationships between ABMs and real-world data. Recently, many studies have addressed the problem of estimating and calibrating ABMs (see Fagiolo et al., 2017, for a recent survey). As stated by Chen et al. (2012), ABMs need to move from stage I, i.e. the capability to grow stylized facts in a qualitative sense, to stage II, where appropriate parameter values are selected according to sound econometric techniques. In those cases where the model is sufficiently simple and well behaved, one can derive a closed form solution for the distributional properties of a specific output of the model, and then estimating the parameters governing such distributions (see e.g. Alfarano et al., 2005, 2006; Boswijk et al., 2007). However, when model complexity prevents analytical solutions, more sophisticated techniques are required. Amilon (2008) estimates a model of financial markets with 15 parameters (with only 2 or 3 agents) using the method of simulated moments[6], reporting high model sensitivity to assumptions made on the noise term and stochastic component of the procedure. Gilli and Winker (2003) and Winker et al. (2007) introduce an algorithm and objective function to estimate exchange-rate models by indirect inference[7], pushing them closer to the properties of real data. Franke (2009) refines on this framework to estimate 6 parameters of an asset pricing model. Franke and Westerhoff (2012) propose a model contest over structural stochastic volatility models, but the models are defined

[5] See e.g. Dosi et al. (2010, 2013, 2015); Caiani et al. (2016a); Assenza et al. (2015) and Dawid et al. (2014a) on business cycle dynamics, Lamperti et al. (2017) on growth, green transitions and climate change, Dawid et al. (2014b) on regional convergence and Leal et al. (2014) on financial markets. The surveys in Fagiolo and Roventini (2012, 2017) provides a more exhaustive list.

[6] The method of simulated moments was introduced as an approach to estimating moment functions when they can not be evaluated directly. See Gilli and Winker (2003); Franke and Westerhoff (2012) for more information on its use in the macro literature.

[7] Note that the method of simulated moments is a form of indirect inference.

by only a few parameters.[8] Finally, Recchioni et al. (2015) use a simple gradient-based procedure for calibration, evaluating performance based on out-of-sample forecast errors.

A parallel stream of research has recently focusing on developing tools to investigate how well ABMs approximate reality (see Marks, 2013; Lamperti, 2017, 2016; Barde, 2016b,a; Guerini and Moneta, 2016). Some of these contributions propose objective functions that replace longitudinal moments within an estimation setting (e.g. the GSL-div introduced in Lamperti, 2017). A common limitation shared by all these methods is the expense of simulating ABMs. As well discussed in Grazzini et al. (2017), simulating the ABM is the most expensive step in calibration, estimation and validation.[9] As an illustrative example, the method proposed in Barde (2016b) requires each Monte Carlo (MC) evaluation to produce time series of 2^{19} periods. Many macroeconomic ABMs might require weeks to perform a single MC exercise of this kind. This might explain why the vast majority of such a literature relies on extremely simple ABMs (with only a few parameters, few agents and no stochastic draws). Conversely, many large macroeconomic ABMs are poorly validated and calibrated, possible revealing underlying computational constraints. New alternative methods must deal with two issues: considerable reduction in computational time and design of appropriate criteria for calibration and validation procedures.

This paper shows that reducing computational time can be achieved in a meaningful way by efficiently training a surrogate model over multiple rounds to approximate the mapping between ABM inputs and the response of the ABM output to a user-defined calibration criterion. Our procedure has some similarities to the one of Dawid et al. (2014b), where penalized splines methods are employed to shortcut parameter exploration and unravel the dynamic effects of policies on the economic variables of interest. However, our method especially focuses on computational efficiency and therefore builds on two pillars: surrogate modelling and intelligent sampling.

Our approach is akin to kriging (also known as Gaussian process regression in the machine learning literature, see Rasmussen and Williams, 2006; Conti and O'Hagan, 2010), which has been introduced as a surrogate modeling approach to facilitate parameter space exploration and sensitivity analyses of ABMs (Salle and Yildizoglu, 2014; Dosi et al., 2016, 2017c,b; Bargigli et al., 2016). This spatial interpolation technique estimates the ABM response over the full parameter space from a finite sample of ABM evaluations to generate the best unbiased *linear* predictor through knowledge of the true variogram or true degree of spatial dependence in the data. In the case of spatially homogeneous data, kriging only requires 30 points to estimate the spatial structure. However, when the spatial distribution of the data is unknown, as is often the case with ABMs, kriging requires specialist knowledge of variography to empirically estimate the spatial dependence of the data. This generally requires a large number of ABM evaluations and extensive exploratory data analysis that increases with the size of the parameter space. Unfortunately, the performance of any kriging model depends on the accuracy of estimating this true variogram, as the empirical variogram asymptotically converges to the true one when the number of ABM evaluations reaches infinity. Our surrogate machine-learning approach should allow to overcome such limitations at negligible computational costs. Let us present

[8] See also Grazzini and Richiardi (2015) and Fabretti (2012) for other applications of the same approach.

[9] The macro-DSGE literature has worked in parallel with the ABM literature in developing techniques to estimate models with many parameters, mostly in the Bayesian tradition (Fernández-Villaverde et al., 2016). However, contrary to ABM modelers, DSGE modelers have rarely faced high computational costs in generating output from their models. The calibration procedure described in this paper might be applied, in principle, to any model involving the production of a storable output. Gains tend to overcome the costs of learning a surrogate when the model is not costless to simulate. For this reason, we believe the present approach has little to no appeal for the DSGE community. Rather, it can contribute to close the gap with the macro-ABM literature in taking models to the data.

it in the the next section.

3 Surrogate Modeling

3.1 Setting specification

This paper proposes an iterative algorithm to efficiently approximate a surrogate model for any ABM using a limited *budget* $B \in \mathbb{N}$ of ABM evaluations. Once this budget is reached, the surrogate model's approximation of the ABM is complete and the surrogate is available to provide a nearly costless approach to predict the model's response.[10]

In all generality, one can represent an agent-based model as a mapping $m : I \to O$ from a set of input parameters I into an output set O. The set of parameters can be conceived as a multidimensional space spanned by the support of each parameter. Usually, the number of parameters go from 1 or 2 to few dozens, as in large macro models. The output set is generally larger, as it corresponds to time-series realizations of a very large number of micro and macro level variables. This rich set of outputs allows a qualitative validation of agent-based models based on their ability to reproduce the statistical properties of empirical data (e.g. non-stationarity of GDP, cross-correlations and relative volatilities of macroeconomic time series), as well as microeconomic distributional characteristics (e.g. distribution of firms' size, of households' income, of assets' returns). Beyond stylized facts, the quantitative validation of an agent-based model also requires the calibration/estimation of the model on a (generally small) set of aggregate variables (e.g. GDP growth rates, inflation and unemployment levels, asset returns etc.).

In practice, such a quantitative calibration consists in the determination of input values such that the output satisfies certain calibration conditions, coming from, e.g., a statistical hypothesis test or the evaluation of a likelihood or loss function. This is in line, for example, with the method of simulated moments (Gilli and Winker, 2003; Franke and Westerhoff, 2012) or the simulated minimum distance approach (Grazzini and Richiardi, 2015). The assessment of a model's output is carried out by computing a specific indicator, which we shall call the *calibration measure*. Two settings are considered:

- **Binary outcome**. The calibration measure might take just two values: 1 if a certain property (or set of properties) on the output is satisfied, or 0 otherwise. For example, one might want to test whether a financial ABM shows excess kurtosis in the distribution of simulated returns or self-sustained GDP growth occur.

- **Real-valued outcome**. The calibration measure is a real-valued number providing a quantitative assessment of a certain property of the model. For example, one might want to compute excess kurtosis of simulated data, or the average GDP growth rate.

Obviously, one would like to find the set of input parameters $x \in I$ such that the calibration measure satisfies certain conditions, which we call *calibration criteria*. To continue with the illustration, consider a user investigating non-normal time series realizations from an ABM. Further, assume that non-normality is measured through negative skew and excess kurtosis, which are then identified as

[10] Notwithstanding its precision, the surrogate remains an approximation of the original ABM. It can be employed to identify those parameter combinations satisfying certain conditions, but further investigations of model behavior around such combinations should be performed using the original ABM.

our calibration measures. Once the ABM generates the time series, both the skew and kurtosis are computed. In the classification outcome setting, the calibration criterion might simply require the presence of both excess kurtosis and negative skew. This reflects the indirect validation approach undertaken in, e.g., Dosi et al. (2015); Caiani et al. (2016b) and Popoyan et al. (2017a). In the real-valued outcome setting, instead, the calibration criterion might involve the comparison of the computed calibration measures with specified thresholds, with the empirical counterparts or with the same quantities computed for other parameter combinations. In particular, in case one chooses a calibration criterion requesting that the distance between empirical and simulated skew and kurtosis should be lower than all previously evaluated points, she would obtain the standard calibration problem of minimizing some loss function over the parameter space. This would mirror exercises like those proposed in Bianchi et al. (2007); Fabretti (2012) or Grazzini et al. (2017).

Then, the following definition identifies positive and negative calibrations.

Definition 1. *A **positive calibration** is a parameter vector $x \in I$ such that model's output satisfies the calibration criterion, whereas one gets a **negative calibration if the opposite occurs**.*

Assume that the calibration criterion takes the form of a simple sign test of the skew and the excess kurtosis. Positive calibrations must show a negative sign on the skew and positive one on the excess kurtosis. All other combinations of skew and kurtosis are negative calibrations. "Positive" and "negative" are what we call *labels* of the points in the parameter space.[11] The objective will be then to find all positive calibrations conditioned on a limited budget B of evaluations.

Remark 1. *Positive calibrations may exist in multiple locations of the parameter space, potentially including those that correspond to economically difficult-to-interpret conditions. ABMs are not designed to mirror every economic scenario, but they must provide reasonable results for those they have been designed for. Ultimately, one should assume that positive calibrations lie along several regions of the parameter space and may not sit solely along regions that are contiguous, or connected to a meaningful economic interpretation. In fact, the most reasonable assumption seems to be that there exist an increasing number of equivalent positive calibrations as the ABM increases in complexity. Finding these areas can be relevant, from the modeller's perspective, to evaluate the model, its reliability and domain of application.*

Remark 2. *The topology of the ABM's response can be characterized by a smooth transition between areas of positive and negative calibrations. Unfortunately, this might also not be the case. Positive calibrations may exist in several regions of the space without much structure. For example, given a combination of parameters of a financial market ABM, the estimated kurtosis of a distribution generated by simulated data may indicate Gaussian returns. This does not guarantee that neighboring points in the parameter space provide the same evidence. The ABM might exhibits "knife-edge" properties, where the response can be described as having a discontinuous and clustered topology (see, e.g., Gualdi et al., 2015; Lamperti, 2016). One of the purposes of the procedure described below is to avoid assumptions on the response surface, which might show either smooth or non-smooth, contiguous or non-contiguous regions.*

[11] We recall that in the machine learning literature supervised learning refers to the problem of making predictions about the label of some data-point using a training labelled dataset. When the training dataset includes non-labelled data we face what is called semi-supervised learning problem. The exercise we perform in this paper falls in the latter category.

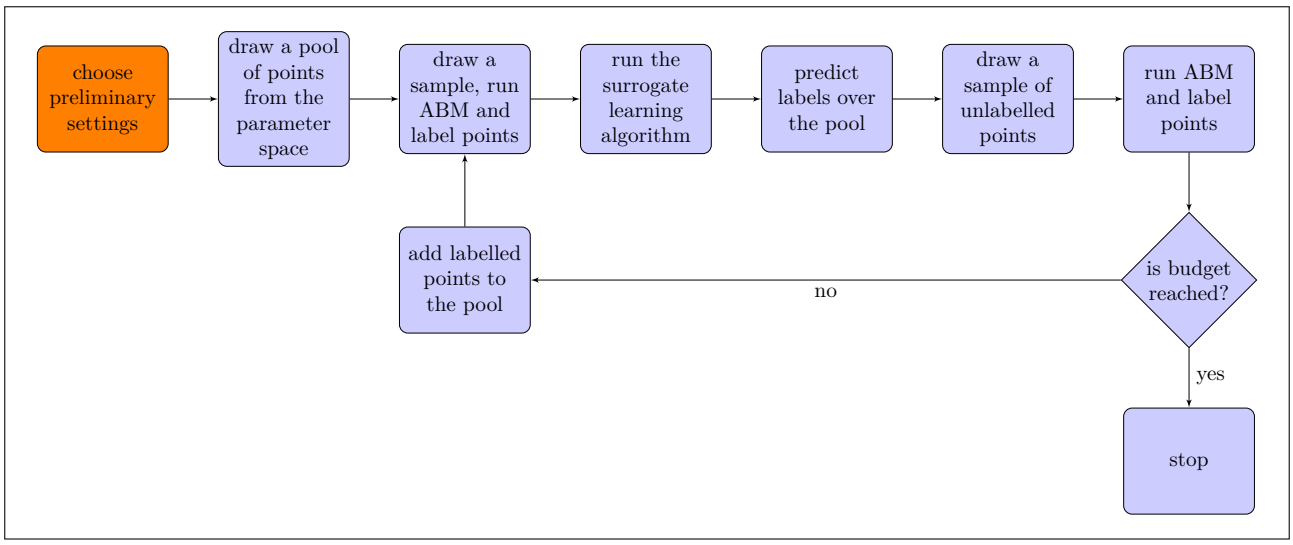


Figure 1: Schematic representation of the proposed procedure to learn a surrogate for an ABM.

3.2 Step-by-step implementation

In the process of finding positive calibrations, it is crucial to drastically reduce the computational time. To do so, we construct our machine learning surrogate following the procedure represented in Figure 1 and further detailed below.

First, the surrogate training procedure requires three preliminary decisions:

1. **Selection of the surrogate algorithm**. The user must choose a machine learning algorithm to act as a surrogate for the original ABM, taking care that the assumptions made by the machine learning model do not force unrealistic assumptions on the response generated over the parameter space.

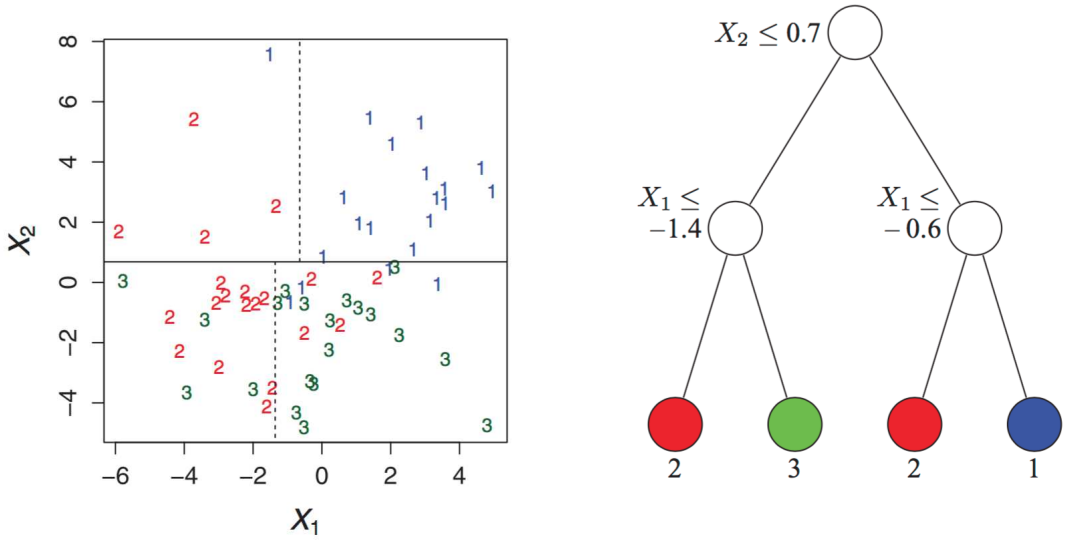
2. **Selection of a fast uniform sampler**. The user must select a sampling procedure to draw samples from the parameters space in order to train the surrogate.

3. **Selection of the surrogate performance measure**: The user must choose a metric to evaluate the performance of the surrogate.

Once these three choices have been made, the procedure proceeds with the following steps.

I step. The process begins by first drawing a relatively large "pool" of parameter combinations, where each combination is a vector containing a value for each parameter, using any standard sampling routine. In particular, we use quasi-random Sobol sampling (Morokoff and Caflisch, 1994). Such a sampling strategy belongs to the class of quasi-Monte Carlo methods and it outperforms other standard approaches, like Monte Carlo and Latin Hypercubes, especially when one has to sample from distributions with an unknown topology as in our setting. Further, standard design of experiments are computationally costly in high dimensional spaces and show little or no advantage over random sampling (Bergstra and Bengio, 2012; Lee et al., 2015). The pool which is constructed at this stage dictates the ability of the algorithm to learn a good surrogate model. As there is a trade-off between the good approximation of the parameter space provided by the pool and the speed to obtain it, we prefer to adopt faster sampler given the high computational cost of running the models.[12]

[12]However, it is entirely reasonable to use a computationally expensive sampling procedure to draw the pool. For example, a large pool of orthogonal Latin hypercube samples can be drawn to populate the pool.



(a) (b)

Figure 2: Partitions (left) and decision tree structure (right) for a classification tree model. Source Loh (2011).

II step. Evaluating the initialization samples consists in running the selected parameter combinations through the ABM and running the output(s) from the ABM through a selected calibration measure and (optional) calibration criterion to determine which of these parameter combinations are positive or negative calibrations.[13]

These two previous steps constitute the initialization phase of our procedure. The following steps define the iterative process.

III step. A meta-model is learned over the initial labeled values in order to build the surrogate.[14] In this paper, we approximate a surrogate by building a set (or "ensemble") of decision trees to fit the labeled parameter combinations.[15] In particular, we rely on XGBoost (Chen and Guestrin, 2016) as our surrogate model. XGBoost implements gradient boosting in a variety of programming languages and is freely available for download from the associated GitHub repository.[16] Gradient boosting consists in a machine learning technique for regression and classification problems. In our setting it produces a statistical model that predicts the label of points in the pool. Such a model is an ensemble of simpler decision trees (see Remark 3), which are aggregated to improve the overall prediction performance. XGBoost builds the model in a stage-wise fashion like other boosting methods do, but it generalizes them by allowing optimization of an arbitrary differentiable loss function. Details about the CARTS and XGBoost are contained in Appendix A.

Remark 3. *Decision trees are designed to efficiently solve classification problems relying on the use of decision paths, which are a set of conditional statements that result in a binning of the data. In the construction of a decision tree, the bin is referred to as the "leaf" of the path or one of the full tree. For example, consider the space spanned by the support of two parameters,* X_1 *and* X_2*,*

[13] Note that as the sample size should contain at least a single positive calibration, we suggest to employ at least 100 samples.

[14] Note there no single algorithm is the best for all types of data. As a consequence, choosing the surrogate algorithm depends on the kind of data one is working with (see e.g. Wolpert, 2002)

[15] The use of decision trees in the analysis of ABMs is not a complete new. See Sridharan and Tesauro (2002) and Dupouët and Yıldızoğlu (2006) for two examples. For additional details on classification and regression trees (CART), please see the remark 3 and Appendix A. For an econometric introduction to CART, see Mullainathan and Spiess (2017).

[16] https://github.com/dmlc/xgboost.

as illustrated in Figure 2a. Assume that the calibration measure allows to identify three different behaviours corresponding to the labels of parameter combinations ("1","2" and "3"). We now want to learn a decision tree model that predicts such behaviours given any vector of parameter values provided as input. One possible decision path leading to the identification of behaviour "2" is: $X_2 \leq 0.7$ *AND* $X_1 \leq -1.4$. *Then, every parameter value satisfying these statements will be predicted as behaviour "2". The same will hold for the following decision path:* $X_2 > 0.7$ *AND* $X_1 < -0.6$. *The core issue here is obviously to find the optimal tree structure. If one evaluates all possible parameter combinations and construct a tree with all possible paths, she could fully characterize the relationship between input parameters pairs and the ABM behavior. Given that constructing such a tree is prohibitive and one has typically a limited budget of ABM evaluations, she must select a subset of parameter combinations to learn a classification model that provides the best representation on a subset of all possible paths. XGBoost provides an intelligent procedure to generate decision paths over a set of trees, and provide the relevant advantage of avoiding any assumption about the structure of the response-surface of the calibration measure. Further, note that in our procedure the classification is simpler than in the example: the surrogate model is used to predict one or two possible labels as a parameter vector can be either a positive or negative calibration.*

IV step. The surrogate model approximated on the set of evaluated samples is used to predict the response over the parameter combinations in the pool, i.e. real-valued responses in the case of a calibration measure and a category in the case of a calibration criterion.

V step. A very small subset of the unlabeled combinations is drawn from the pool and evaluated in the ABM, labeled according to the application of the calibration criterion and added to the set of labeled combinations within the pool. Two issues need to be addressed at this stage. First, how many new points should be sampled and, second, how to select them. In line with the results in Ross et al. (2011), the total number of additional parameters vectors that are drawn at this stage is the logarithm of the budget. Concerning new data points selection, the algorithm randomly selects parameter combinations among those points having positive predictions. The procedure incrementally increases true positives, while reducing false ones. If there are no new positive predictions in the current round, new points are added to the set of labeled combinations. In the absence of probabilities that predict a positive label, we use the so-called uncertainty sampling (Lewis and Gale, 1994). Such technique relies on the entropy of the distribution of existing labels in order to increase the sampling frequency of parameter combinations that are difficult to label correctly by the surrogate. In this way we reduce the discrepancy, in terms of sampling, between the regions that contain a manifold of interest, which are usually sampled by our algorithm, and those where the surrogate tend to fail, which are more informative from a learning perspective (Zhu, 2005; Goldberg et al., 2011).

The previous three steps are repeated until the budget of ABM evaluations is reached.[17]

Remark 4. *Standard design of experiments are computationally costly to compute and show little or no advantage over random sampling (Bergstra and Bengio, 2012; Lee et al., 2015). Alternatively, iterative sampling approaches from the machine learning literature exploit the information gained from sample to sample and offer a variety of ways to choose samples over multiple rounds to improve*

[17] Appendix A contains the pseudo-code of the algorithm together with additional technical details. The online supplementary material provides the Python functions that allow the user to replicate the exercises presented in this paper and the following repository, `https://github.com/amirsani/online_surrogate_modeling`, contains a working example and a comparison of our algorithm with kriging meta-modelling.

sampling performance (Settles, 2010; Cohn et al., 1994).[18] *Here, we exploit the information gained through iterative sampling and use it to intelligently direct the selection of new data points. Ross et al. (2011) provides convergence rates for aggregating samples over multiple rounds without making any assumption on the distribution that generated the data. As our approach performs the same iterative aggregation of labeled data over multiple rounds without any assumptions on the underlying distribution, we can use their result - which suggest to sample log(budget) points - to provide a guideline on how many parameter combinations to label at each round. Of course, it is entirely reasonable to use different approaches. The statement is only meant to provide users with a guideline heuristic.*

Remark 5. *The proposed approach share with iterative Monte Carlo methods (iMC, see Doucet et al., 2000 for an example) the feature of recursive sampling. In particular it aggregates informative samples for accurate predictive inference by sampling predicted positive calibrations from a non-stationary approximation of the distribution of positive and negative calibrations in the parameter space. On the other side iMC methods, such as the Metropolis Hastings algorithm (Metropolis et al., 1953), iteratively and directly sample the distribution at random, traversing along an assumed Markov chain until enough samples have been collected to converge to the stationary distribution generating the samples (Chib and Greenberg, 1995). A series of differences emerges. First, the proposed approach does not assume that a Markov chain underlies the sequence of samples. Second, it does not assume that the approximated distribution is stationary over rounds, as each additional set of labelled samples changes the approximation in each round. Third, no statistical assumptions are made on the parameter distribution. Finally, our approach relies on a pool of unlabelled combinations from the parameter space as a proxy for the full population; it first predicts the values of unlabelled combinations in the pool and then samples from the predictions, conditioned on a positive prediction. On the contrary, iMC methods directly sample and evaluate the samples in each round.*

3.3 Surrogate performance and its evaluation

During the surrogate learning stage of our algorithm (step III in Section 3.2), the meta-model is evaluated and optimized. In practice, this means that XGBoost tries to find the best CART model for the data it processes in each round. To do so, it needs a performance (or loss) measure to be maximised (or minimized).

As outlined above the paper considers two settings: a binary outcome setting, where any test that compares simulated output with real data is considered (e.g. a non-parametric test on distribution equality); and a real-valued outcome setting, where any quantifiable feature the model might generate is considered (e.g. difference between growth rates in real and simulated data). In the case of a binary response, the performance objective is to maximize the standard classification accuracy measure F_1 (Fawcett, 2006):

$$F_1 = \frac{2 \cdot \text{true positives}}{2 \cdot \text{true positives} + \text{false positives} + \text{false negatives}}, \tag{1}$$

which is bounded between 0 and 1. In the case of a real-valued response, the performance objective is to minimize the mean squared error (MSE):

$$MSE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}, \tag{2}$$

[18] Some examples include uncertainty sampling (Lewis and Gale, 1994), query by committee (Seung et al., 1992; Balcan et al., 2006), and error minimization (Roy and McCallum, 2001; Timothé and Olivier, 2015).

where $\hat{y}_i$ indicates the surrogate's prediction, y_i corresponds to the true value, and N is the size of the sample of learning points, i.e. the points in the pool that have been evaluated through the ABM. The use of the MSE is in line with the loss function employed in recent calibration exercises (see e.g. Recchioni et al., 2015).

Once the budget has been reached and the surrogate approximated, an out-of-sample set of points is used to test the performance of our meta-model in finding positive calibrations. In both settings, we rely on the so-called "True Positive Rate" (TPR, see Fawcett, 2006):

$$TPR = \frac{\text{number of correctly predicted positives}}{\text{number of positives in the pool}}, \tag{3}$$

which measures the proportion of calibrations correctly predicted by the surrogate against the number of true positive calibrations. In practice, the TRP cannot be measured without evaluating all the possible parameters combinations. However, we decided to present it in our simulated scenarios because it is intuitive, easy to compute and provides a nice summary statistic to validate the performance of our surrogate.

3.4 Parameter importance

The algorithm provides an intuitive procedure to assess the importance of each parameter in explaining the variance of the data by counting the relative number of times a parameter was "split-on" in the CART ensemble (for details see e.g. Archer and Kimes, 2008; Louppe et al., 2013; Breiman, 2001). As each tree is constructed according to an optimized splitting of the possible values for a specific parameter vector, and it is increasingly focusing on difficult-to-predict samples, splits dictate the relative importance of parameters in discriminating the output conditions of the ABM. Accordingly, the relative number of splits over a specific parameter provides a quantitative assessment of the sensitivity of the surrogate to the parameter and the importance of that parameter to the user-specified conditions. As a consequence, one can rank rank model's parameters on the basis of their importance in producing a behavior of the model that satisfies the calibration criterion.

However, the relative feature importance does not represent a sensitivity analysis of the parameters and does not provide the direction of influence. Indeed, given the non-linear response of ABMs to changes in parameters, many additional samples would be necessary to approximate the first, second and total order variation and direction of sensitivities. Given the computational cost of such an exercise, this is often ignored in the literature and replaced by simulated sensitivities obtained from a linear kriging model. The result is an approximate sensitivity which could be far from the one. We then believe that before moving to sensitivity analysis, one must extensively test the performance of any meta-model in approximating the true ABM. For instance, we compare our procedure with kriging in Appendix B and our results suggest that our proposed machine learning surrogate is more accurate in approximating the true model than kriging. We intentionally leave the issue of incorporating sensitivity analysis in our approach to future research.

4 Application I: The Brock and Hommes model

In their seminal contribution, Brock and Hommes (1998) develop an asset pricing model (referred here as B&H), where an heterogeneous population of agents trade a generic asset according to different strategies (fundamentalist, chartists, etc.). In what follow, we first briefly introduce the model (cf.

Section 4.1). We then report the empirical setting (see Section 4.2) and the results of our machine learning calibration and exploration exercise (cf. Section 4.3). We recall that the seed of the pseudo-random number generator is fixed and kept constant across runs of the model over different parameter vectors.

4.1 The B&H asset pricing model

There is a population of N traders that can invest either in a risk free asset, which is perfectly elastically supplied at a gross return $R = (1 + r) > 1$, or in a risky one, which pays an uncertain dividend y and has a price denoted by p. Wealth dynamics is given by

$$W_{t+1} = RW_t + (p_{t+1} + y_{t+1} - Rp_t)z_t, \tag{4}$$

where p_{t+1} and y_{t+1} are random variables and z_t is the number of the shares of the risky asset bought at time t. Specifically, an agent's wealth at time $t+1$ equals the sum of previously accumulated wealth multiplied by its gross return and the market gain obtained in that period, where the latter is given by the dividend y_{t+1} plus the capital gain $(p_{t+1} - Rp_t)$ for each share.

Traders are heterogeneous in terms of their expectations about future prices and dividends and are assumed to be myopic mean-variance maximizers. However, as information about past prices and dividends is publicly available in the market, agents can apply conditional expected value E_t, and variance V_t. The demand for share $z_{h,t}$ of agents with expectations of type h is computed solving:

$$\max_{z_{h,t}} \left\{ E_{h,t}(W_{t+1}) - \frac{\nu}{2} V_{h,t}(W_{t+1}) \right\}, \tag{5}$$

which in turns implies

$$z_{h,t} = E_{h,t}(p_{t+1} + y_{t+1} - Rp_t)/(\nu\sigma^2), \tag{6}$$

where ν controls for agents' risk aversion and σ indicates the conditional volatility, assumed to be equal across traders and constant over time. In case of zero supply of outside shares and different trader types, the market equilibrium equation can be written as:

$$Rp_t = \sum n_{h,t} E_{h,t}(p_{t+1} + y_{t+1}), \tag{7}$$

where $n_{h,t}$ denotes the share that traders of type h hold at time t. In presence of homogeneous traders, perfect information and rational expectations, one can derive the no-arbitrage market equilibrium condition:

$$Rp_t^* = E_t(p_{t+1}^* + y_{t+1}), \tag{8}$$

where the expectation is conditional on all histories of prices and dividends up to time t and where p^* indicates the fundamental price. In case dividends are independent and identically distributed over time with constant mean, equation (8) has a unique solution where the fundamental price is constant and equal to $p^* = E(y_t)/(R - 1)$. In what follows, we will express prices as deviations from the fundamental price, i.e. $x_t = p_t - p_t^*$.

At the beginning of each trading period $t = \{1, 2, ..., T\}$, agents form expectations about future prices and dividends. Agents are heterogeneous in their forecasts. More specifically, investors believe that, in a heterogeneous world, prices may deviate from the fundamental value by some function $f_h(\cdot)$ depending upon past deviations from the fundamental price. Accordingly, the beliefs about p_{t+1} and

y_{t+1} of agents of type h evolve according to:

$$E_{h,t}(p_{t+1} + y_{t+1}) = E_t(p^*_{t+1}) + f_h(x_{t-1}, ..., x_{t-L}). \tag{9}$$

Many forecasting strategies specifying different trading behaviours and attitudes have been studied in the economic literature, (see e.g. Banerjee, 1992; Brock and Hommes, 1997; Lux and Marchesi, 2000; Chiarella et al., 2009). Brock and Hommes (1998) adopt a simple linear representation of beliefs:

$$f_{h,t} = g_h x_{t-1} + b_h, \tag{10}$$

where g_h is the trend component and b_h the bias of trader type h towards a particular value of the price. If $b_h \neq 0$, the agent h can be either a pure trend chaser if $g_h > 0$ (strong trend chaser if $g > R$), or a contrarian if $g < 0$ (strong contrarian if $g < R$). If $g_h = 0$, the agent of type h does not believe in any trending movement but is just influenced by the bias. In the special case when both g_h and b_h are equal to zero, the agent is a "fundamentalists", i.e. she believes that prices return to their fundamental value. Agents can also be fully rational, with $f_{rational,t} = x_{t+1}$. In such a case, they have perfect foresight but, they must pay a cost C.[19]

In our application, we use a simple model with only two types of agents, whose behaviours vary according to the choice of trend components, biases and perfect forecasting costs. Combining equations (7), (9) and (10), one can derive the following equilibrium condition:

$$Rx_t = n_{1,t} f_{1,t} + n_{2,t} f_{2,t}, \tag{11}$$

which allows to compute the price of the risky asset (in deviation from the fundamental) at time t.

Traders switch among different strategies according to the their evolving profitability. More specifically, each strategy h is associated with a fitness measure of the form:

$$U_{h,t} = (p_t + y_t - Rp_{t-1}) z_{h,t} - C_h + \omega U_{h,t-1} \tag{12}$$

where $\omega \in [0, 1]$ is a weight attributed to past profits. At the beginning of each period, agents reassess the profitability of their trading strategy with respect to the others. The probability that an agent choose strategy h is given by:

$$n_{h,t} = \frac{\exp(\beta U_{h,t})}{\sum_h \exp(\beta U_{h,t})}, \tag{13}$$

where the parameter $\beta \in [0, +\infty)$ captures traders' intensity of choice. According to equation 13, successful strategies gain an increasing number of followers. In addition, the algorithm introduces a certain amount of randomness, as less profitable strategies may still be chosen by traders. In this way, the model captures imperfect information and agents' bounded rationality. Moreover, the system can never be stacked in an equilibrium where all traders adopt the same strategy.

4.2 Experimental design and empirical setting

Despite the model being relatively simple, different contributions have tried to match the statistical properties of its output with those observed in real financial markets (Boswijk et al., 2007; Recchioni

[19] In our experiments we allow for the possibility that a positive cost might be by paid also by non-rational traders. This mirrors the fact that some trader might want to buy additional information, which they might not be able to use (due e.g. to computational mistakes).

Table 1: Parameters and explored ranges in the Brock and Hommes model.

Parameter	Brief description	Theoretical support	Explored range
	Brock and Hommes Model		
β	intensity of choice	$[0; +\infty)$	$[0.0; 10.0]$
n_1	initial share of type 1 traders	$[0; 1]$	0.5
b_1	bias of type 1 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
b_2	bias of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
g_1	trend component of type 1 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
g_2	trend component of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
C	cost of obtaining type 1 forecasts	$[0; +\infty)$	$[0.0; 5.0]$
ω	weight to past profits	$[0.0, 1.0]$	$[0.0; 1.0]$
σ	asset volatility	$(0; +\infty)$	$(0.0; 1.0]$
ν	attitude towards risk	$[0; +\infty]$	$[0; 100]$
r	risk-free return	$(1; +\infty)$	$[1.01, 1.1]$
T_{BH}	number of periods	$\mathcal{N}$	500

et al., 2015; Lamperti, 2016; Kukacka and Barunik, 2016). This makes the model an ideal test case for our surrogate: it is relatively cheap in terms of computational needs, it offers a reasonably large parameter space and it has been extensively studied in the literature.

There are 12 free parameters (Table 1) whose values are to be determined trough calibration.[20] The ranges for parameters' values have been identified relying on both economic reasoning and previous experiments on the model. However, their selection is ultimately a user specific decision. Our procedure allow to deal with large parameter spaces, thus minimizing the constraints face by modellers. In what follows, we refer to the parameter space spanned by the intervals specified in the last column of Table 1. Naturally, it can be further expanded or reduced according to the user's needs and the available budget.

Let us now consider the conditions identifying positive calibrations. As already discussed above, any feature of model's output can be employed to express such conditions. According to Section 3 two types of calibration criteria are considered, giving respectively binary and real-valued outcomes. In the binary outcome case, we employ a two-sample Kolmogorov-Smirnov (KS) test between the distribution of logarithmic returns obtained from the numerical simulation of the model and the one obtained from real stock market data.[21] More specifically, we rely on daily adjusted closing prices for the S&P 500 going from December 09, 2013 to December 07, 2015, for a total of 502 observations, and we compute the following test statistic:[22]

$$D_{RW,S} = \sup_r |F_{RW}(r) - F_S(r)|, \tag{14}$$

where r indicate logarithmic returns and F_{RW} and F_S are the empirical distribution functions of the real world (RW) and simulated (S) samples respectively. Then, in a real-valued outcome setting, we use the p-value of the KS test, $P(D > D_{RW,S})$, as an expression of model's fit with the data. In particular, the higher the p-value of the test, the more difficult to reject the null and the larger the

[20] We underline that the dimension of the parameter space is in line or even larger that in recent studies on ABM meta-modelling (see e.g. Salle and Yildizoglu, 2014; Bargigli et al., 2016).

[21] Let p_t and p_{t-1} be the prices of an asset at two subsequent time steps. The logarithmic return from $t-1$ to t is given by $r_t = \log(p_t/p_{t-1}) \simeq (p_t - p_{t-1})/p_{t-1}$.

[22] The data have been obtained from Yahoo Finance: https://finance.yahoo.com/quote/%5EGSPC/history. The test is passed if the null hypothesis "equality of the distributions" is not rejected at 5% confidence level.

fit with the data. We also consider an equivalent condition for the binary outcome: predicted labels above 5% indicate positive calibrations. The choice is made on purpose: using equivalent conditions allows to compare the binary and real-valued outcome in terms of precision (ability to identify true calibrations) and computational time (in the real-valued scenario there is more information to be processed.)

We train the surrogate 100 times over 10 different budgets of 250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250, 2500 labelled parameter combinations, with a fixed random seed[23], and evaluate it on 100000 unlabelled points. Having a large number of out-of-sample, unlabelled, possibly well-spread points is fundamental to evaluate the performance of the meta-model. We use a larger evaluation set than any other meta-modelling contribution we are aware of (see, for instance, Salle and Yildizoglu, 2014; Dosi et al., 2017c; Bargigli et al., 2016).

4.3 Results

In Figure 3, we show the parameter importance results for the Brock and Hommes (B&H) model. We find that the most relevant parameters to fit the empirical distribution of returns observed in the SP500 are those characterizing traders' attitude towards the trend (g_1 and g_2) and, secondly, their bias (b_1 and b_2). This result is in line with recent findings by Recchioni et al. (2015) and Lamperti (2016) obtained using the same model. Moreover, the intensity of choice parameter (β, cf. Section 4), which is of crucial importance in the original model developed by Brock and Hommes (1998), does not appear to be particularly relevant in determining the fit of the model with the data if compared to other behavioural parameters (at least within the range expressed by Table 1,).[24] Also traders' risk attitude (α) and the weight associated to past profits (ω) are relatively unimportant to shape the empirical performance of the model.

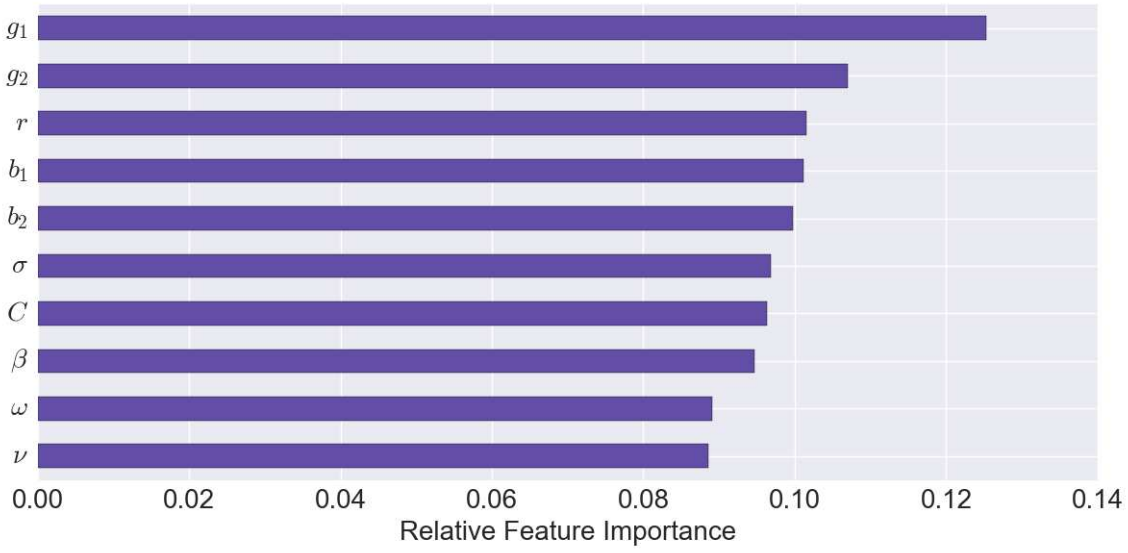


Figure 3: Importance of each parameter (feature) in shaping behaviour of the Brock and Hommes model according to the specified conditions (i.e. equality between distributions of simulated and real returns). As noted in Section 3.4, this chart demonstrates the relative rank-based importance for each parameter.

[23]Note that additional seeds could be used to generate a Monte-Carlo evaluation, as demonstrated in Section 5.4, but it is also possible to use the block-free Bootstrap estimation procedure proposed in Sani et al. (2015).

[24]See also Boswijk et al. (2007) where the authors estimate the B&H model on the SP500 and, in many exercises, find the switching parameter not to be significant.

Let us now consider the behaviour of the surrogate. As outlined in Section 3.3, we run a series of exercises where the surrogate is employed to explore the behaviour of the model over the parameter space and filter out positive calibrations matching the distribution of real stock-market returns. Figure 4 collects the results and show the performance of the surrogate in the two proposed settings (binary and real-valued outcome) for a variable budget size. Within the binary outcome exercise, the F_1-score is steadily increasing with the size of the training sample and it reaches a peak value of around 0.80 when 2500 points are employed (cf. Figure 4a). In other words, the average between the share of true positive calibrations and the share of positive calibrations the surrogate correctly predicts is 0.8. Taken into consideration the upper bound of 1 and various practical applications (e.g. Petrovic et al., 2011; Cireşan et al., 2013), we consider the former result satisfying. However, such a classification performance should be evaluated in view of the surrogate's *searching ability*, which is reported in Figure 4c and indicates the share of total positive calibration that the surrogate is able to find. Specifically, we find that around 75% of the positive calibrations present in the large set of out-of-sample points are found.

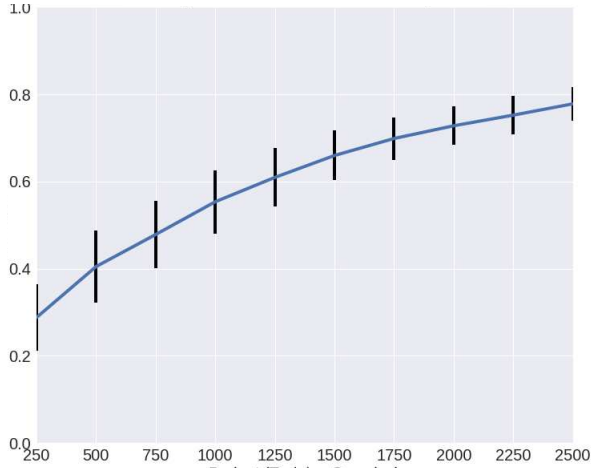
Obviously, the surrogate's performance worsens as the training sample size is reduced. However, once we move to the real-valued setting, where the surrogate is learnt using a continuous variable (containing more information about model's behaviour), its performance is remarkably higher. Indeed, even when the sample size of the training points is particularly low (500), the True Positive Ratio (TPR) is steadily around 70%, and it reaches almost 95% (on average) when 2500 parameter vectors are employed (see Figure 4d).[25]

Timing results are reported according to the average seconds required for a single compute core to complete the specific task 100 times. Specifically, these tasks are: building the surrogate (green line, labeled as âĂIJSurrogateâĂİ), predict labels using the surrogate (blue line, labeled as âĂIJPredictionâĂİ) and evaluate the true label running the ABM and evaluating its output (red line, labeled as âĂIJABMâĂİ). First, we notice that the increase in performance from classification (see Figure 4e) to regression (see Figure 4f) requires roughly 3X the modelling time and a nearly equivalent prediction time. Given such negligible computational costs, our approach facilitates a nearly costless exploration of the parameter space, delivering good results in terms of F_1-score, TPR and MSE. The time savings in comparison to running the original ABM are substantial. In this exercise over a set of 10000 out-of-sample points, the surrogate is 500X faster on average in prediction. Note also that the learned surrogate is reusable on any number of out-of-sample parameter combinations, without the need for additional training. Further, we remark that computational gains are expected to be larger as more complex and expensive-to-simulate models are used. The next section goes in this direction.

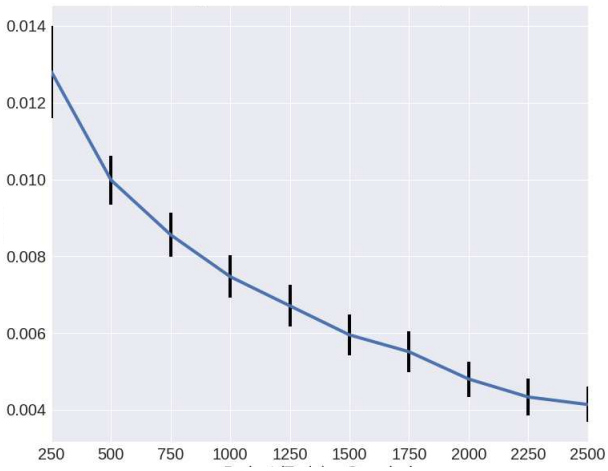
5 Application II: The Islands model

In the "Island" growth model (Fagiolo and Dosi, 2003), a population of heterogeneous firms locally interact discovering and diffusing new technologies, which ultimately lead to the emergence (or not) of endogenous growth. After having presented the model (Section 5.1), we describe the empirical setting (see Section 5.2) and the results of the machine learning calibration and exploration exercises (cf. Section 5.3). We recall that the seed of the pseudo-random number generator is fixed and kept

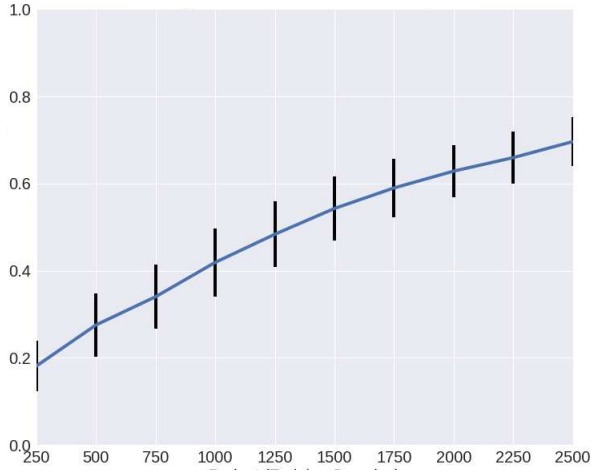
[25] We notice that the observation of the relationship between TRP and budget size might suggest a stopping rule in determining a reasonable budget size for the model. For example, a rule of thumb like: âĂIJif the marginal performance gain from increasing the budget size is increasing, then keep enlarging the budget; stop otherwiseâĂİ may be a valuable option.



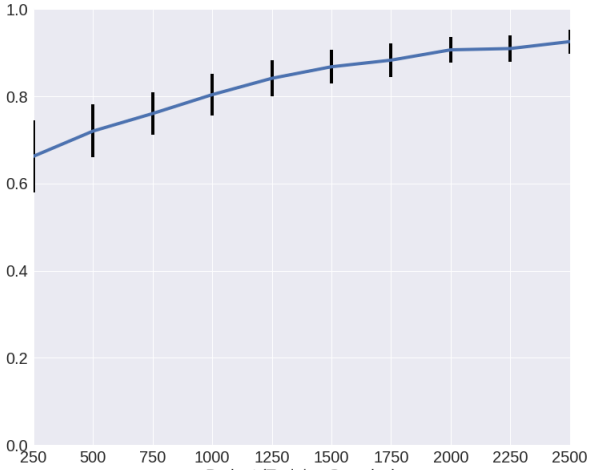
(a) Binary-outcome: F1 Score



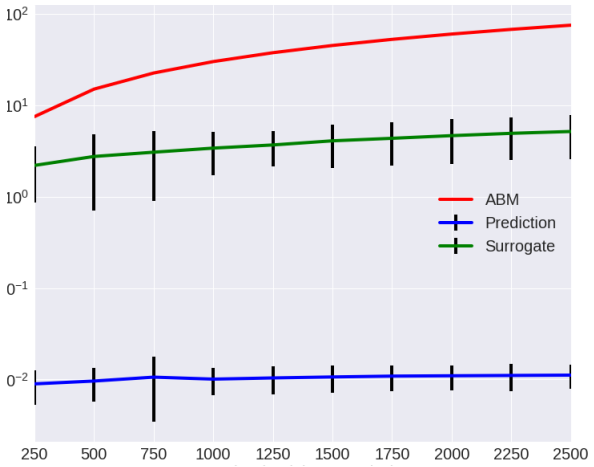
(b) Real-valued outcome: Mean Squared Error



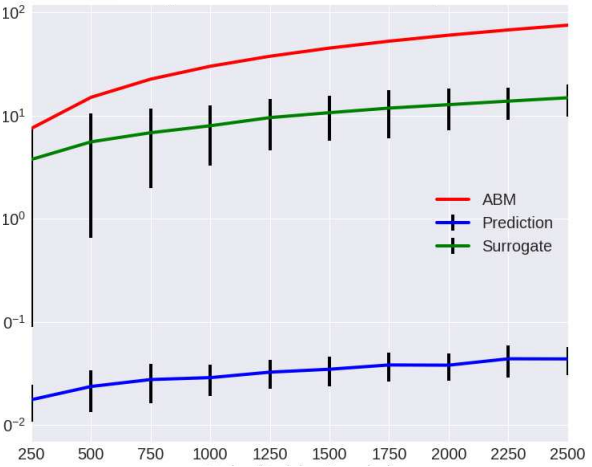
(c) Binary-outcome: True Positive Rate



(d) Real-valued outcome: True Positive Rate



(e) Binary-outcome: Computation Time (seconds)



(f) Real-valued outcome: Computation Time (seconds)

Figure 4: Brock and Hommes surrogate modelling performance averaged over a pool of 10000 parametrizations. Black vertical lines indicate 95% confidence intervals on 100 repeated and independent experiments. Budget size is on the X-axis. Y-axis in log scale for panels 4e and 4f.

constant across runs of the model over different parameter vectors.

5.1 The Island growth model

A fixed population of heterogeneous firms ($I = 1, 2, ..., N$) explore an unknown technological space ("the sea"), punctuated by islands (indexed by $j = 1, 2, ...$) representing new technologies. The technological space is represented by a 2-dimensional, infinite, regular lattice endowed with the Manhattan metrics d_1. The probability that each node (x, y) is an island is equal to $p(x, y) = \pi$. There is only one homogeneous good, which can be "mined" from any island. Each island is characterized by a productivity coefficient $s_j = s(x, y) > 0$. The production of agent i on island j having coordinates (x_j, y_j) is equal to:

$$Q_{i,t} = s(x_j, y_j)[m_t(x_j, Y_j)]^{\alpha-1}, \tag{15}$$

where $\alpha \geq 1$ and $m_t(x_j, y_j)$ indicates the total number of miners working on j at time t. The GDP of the economy is simply obtained summing up the production of each island.

Each agent can choose to be a *miner* and produce an homogeneous final good in her current island, to become an *explorer* and search for new islands (i.e. technologies), or to be an *imitator* and sail towards a known island. In each time step, miners can decide to become explorer with probability $\epsilon > 0$. In that case, the agent leaves the island and "sails" around until another (possibly still unknown) island is discovered. During the search, explorers are not able to extract any output and randomly move in the lattice. When a new island (technology) is discovered, its productivity is given by:

$$s_{j^{\text{new}}} = (1 + W)\{[|x_{j^{\text{new}}}| + |y_{j^{\text{new}}}|] + \varphi Q_i + \omega\} \tag{16}$$

where W is a Poisson distributed random variable with mean $\lambda > 0$, ω is a uniformly distributed random variable with zero mean and unitary variance, φ is a constant between zero and one and, finally, Q_i is the output memory of agent i. Therefore, the initial productivity of a newly discovered island depends on four factors (see Dosi, 1988): (i) its distance from the origin; (ii) cumulative learning effects (ϕ); (iii) a random variable W capturing radical innovations (i.e. changes in technological paradigms); (iv) a stochastic i.i.d. zero-mean noise controlling for high-probability low-jumps (i.e. incremental innovations).

Miners can also decide to imitate currently available technologies by taking advantage of informational spill-overs stemming from more productive islands located in their technological neighbourhoods. More specifically, agents mining on any colonized island deliver a signal, which is instantaneously spread in the system. Other agents in the lattice receive the signal with probability:

$$w_t(x_j, y_j; x, y) = \frac{m_t(x_j, y_j)}{m_t} \exp\{-\rho[|x - x_j| + |y - y_j|]\}, \tag{17}$$

which depends on the magnitude of technology gap as well as on the physical distance between two islands ($\rho > 0$). Agent i chooses the strongest signal and become an imitator sealing to island according to the shortest possible path. Once the imitated island is reached, the imitator will start mining again.

The model shows that the very possibility of notionally unlimited (albeit unpredictable) technological opportunities is a necessary condition for the emergence of endogenous exponential growth. Indeed, self-sustained growth is achieved whenever technological opportunities (captured by both the density of islands π and the likelihood of radical innovations λ), path-dependency (i.e. the fraction of idiosyncratic knowledge, φ, agents carry over to newly discovered technologies), and spreading intensity in the information diffusion process (ρ), are beyond some minimum thresholds (Fagiolo and Dosi, 2003). Moreover, the system endogenously generate exponential growth if the trade-off between

Table 2: Parameters and explored ranges in the Island model.

Parameter	Brief description	Theoretical support	Explored range
	Islands Model		
ρ	degree of locality in the diffusion of knowledge	$[0, +\infty)$	$[0; 10]$
λ	mean of Poisson r.v. - jumps in technology	$[0; +\infty)$	1
α	productivity of labour in extraction	$[0, +\infty)$	$[0.8; 2]$
φ	cumulative learning effect	$[0, 1]$	$[0.0; 1.0]$
π	probability of finding a new island	$[0.0, 1.0]$	$[0.0; 1.0]$
ϵ	willingness to explore	$[0, 1]$	$[0.0; 1.0]$
m_0	initial number of agents in each island	$[2, +\infty)$	50
T_{IS}	number of periods	$\mathcal{N}$	1000

exploration and exploitation is solved, i.e. if the ecology of agents find the right balance between searching for new technologies and mastering the available ones.

5.2 Experimental design and empirical setting

The Island model employs eight input parameters to generate a wide array of growth dynamics. We report the parameters, their theoretical support and the explored range in Table 2. We kept the number of firms fixed (and equal to 50) to study what happens to the same economic system, when the parameters linked to behavioural rules are changed.[26]

Similarly to section 4.2, we characterize a binary outcome and a real-valued outcome setting. In the first case, the surrogate is learnt using a binary target variable y taking value 1 if a user-defined specific set of conditions is satisfied and zero otherwise. More specifically, we define two conditions characterizing the GDP time series generated by the model. The first condition requires the model to generate self-sustained pattern of output growth. Given the long-run average growth rate of the economy (AGR):

$$AGR = \frac{\log(GDP_T) - \log(GDP_1)}{T - 1}, \tag{18}$$

sustained growth emerges if $AGR > 2\%$.

The second condition aims at capturing the presence of fat tails in the output growth-rate distributions. This empirical regularities, which suggest that deep downturns coexist with mild fluctuations has been found in both OECD (Fagiolo et al., 2008) and developing countries (Castaldi and Dosi, 2009; Lamperti and Mattei, 2016). More specifically, we fit a symmetric exponential power distribution (see Subbotin, 1923; Bottazzi and Secchi, 2006) , whose functional form reads:

$$f(x) = \frac{1}{2ab^{\frac{1}{b}}\Gamma(1 + \frac{1}{b})} e^{-\frac{1}{b}|\frac{x-\mu}{a}|^b} \tag{19}$$

where a controls for the standard deviation, b for the shape of the distribution and μ represents the mean. As b gets smaller, the tails become fatter. In particular, when $b = 2$ the distribution reduces to a Gaussian one, while for $b = 1$ the density is Laplacian. We say that the output growth-rate distribution exhibits fat tails if $b \leq 1$. Note that there is a hierarchy in the conditions we have just defined: only those parametrizations satisfying the first one ($AGR > 2\%$) are retained as candidates for positive calibrations and further investigated with respect to the second condition. In the real-

[26] Note that the Island model does not exhibit scale effects: the results generated by the model does not depend on the number of agents in the system (Fagiolo and Dosi, 2003).

valued outcome case, instead, we just focus on shape of growth rates distribution. In particular, we our target variable is the estimated b of the symmetric power exponential distribution and a positive calibration is found if $b > 1$.[27] Again, the choice of the condition to be satisfied ensures (partial, in this case) consistency between the two settings.

We train the surrogate as we did with the B&H model, but given the higher computational complexity of the Island model, we reduce the number of unlabelled points to 10000.[28]

5.3 Results

As for the Brock and Hommes model, we start our analysis reporting the relative importance for all the parameters characterizing the Island model (figure 5). We find that all the parameters of the model linked to production, innovation and imitation appear to be relevant for the emergence of sustained economic growth.

The surrogate's performances is presented in Figure 6, where the first column of the plots refers to the binary outcome setting, while the second one to the real-valued one. The F_1-score displays relatively high values even for low training sample sizes (250 and 500) pointing to a good classification performance of the surrogate (see Figure 6a). However, it quickly saturates, reaching a plateau around 0.8. Conversely, in the real-valued setting, the surrogate's performance keeps increasing with the training sample size, and it displays remarkably low values of MSE when more than 1000 points are employed (cf. Figure 6b).

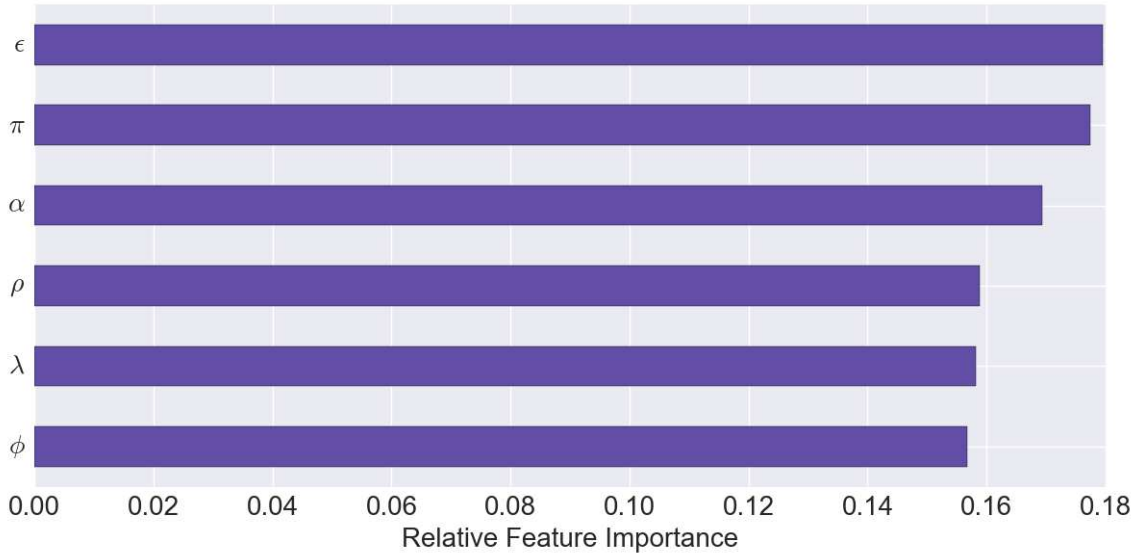
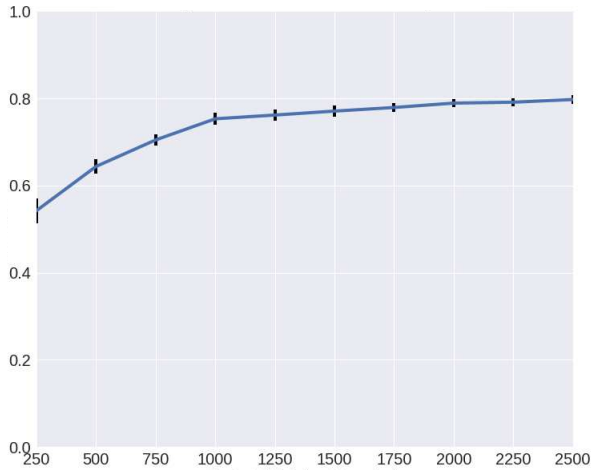


Figure 5: Importance of each parameter (feature) in shaping behaviour of the Islands model according to the specified conditions (sustained growth and fat tails). As noted in Section 3.4, this chart demonstrates the relative rank-based importance for each parameter.

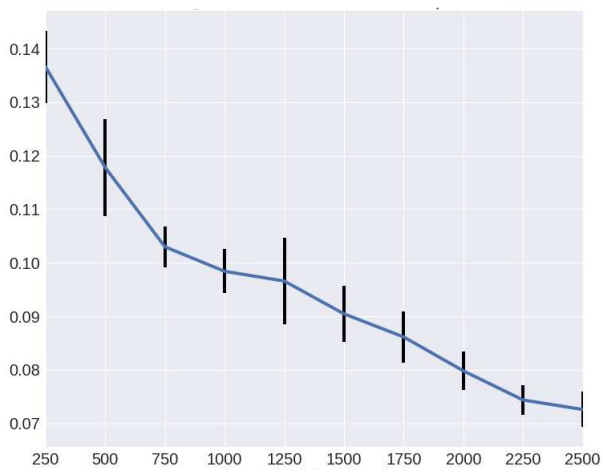
In both settings, the searching ability of the surrogate behaves in a similar way: the TRP steadily increases with the training sample size (cf. Figures 6c and 6d). In absolute terms, the real-valued setting delivers much better results than the binary one, as for the Brock and Hommes model (section 4.3). In particular, the largest true positive ratio reaches 0.9 for the real-valued case and 0.8 for

[27] In the real-valued outcome setting our exercise is comparable to those performed in Dosi et al. (2017c), where the same distribution and parameters are used in a model of industrial dynamics.

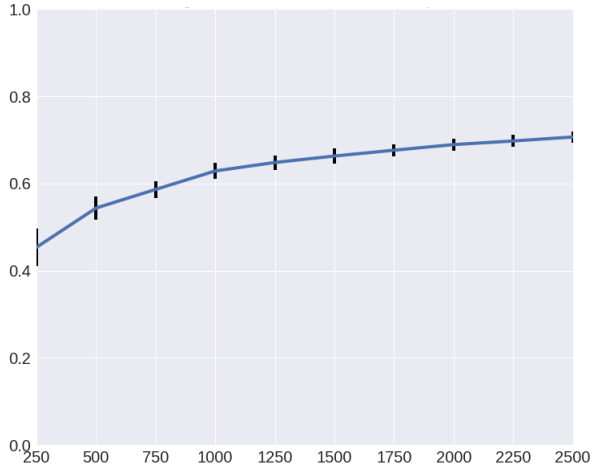
[28] This choice is motivated by the fact that we need to run the model on the out-of-sample points in order to evaluate the surrogate.



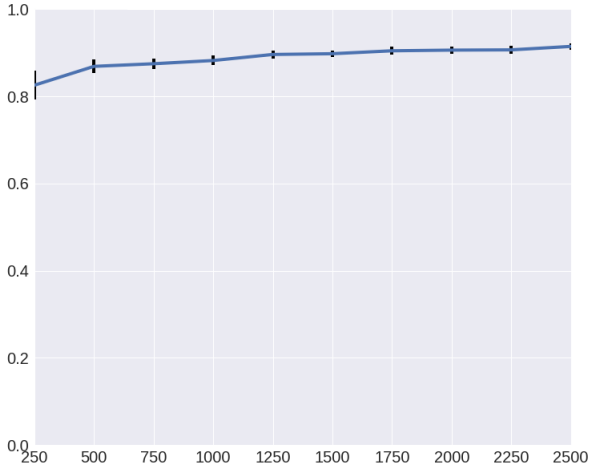
(a) Binary-outcome: F1 Score



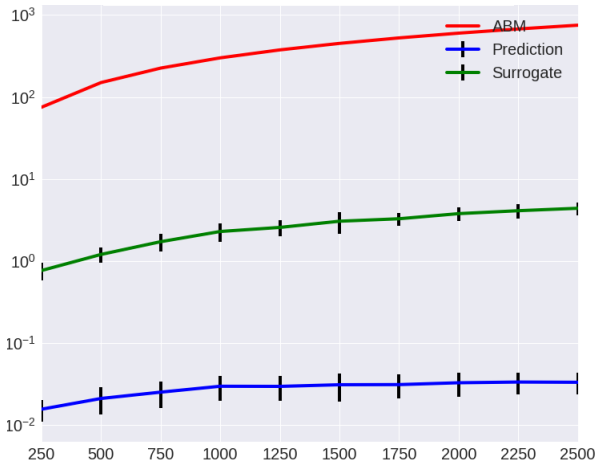
(b) Real-valued outcome: Mean Squared Error



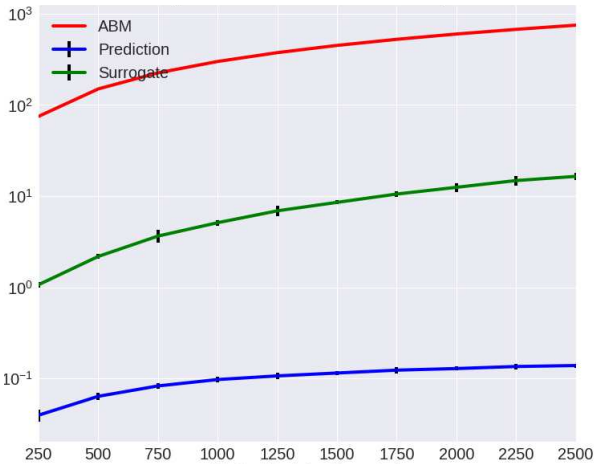
(c) Binary-outcome: True Positive Rate



(d) Real-valued outcome: True Positive Rate



(e) Binary-outcome: Computation Time (seconds)



(f) Real-valued outcome: Computation Time (seconds)

Figure 6: Islands surrogate modelling performance versus budget size averaged over a pool of 10000 parametrizations. Black vertical lines indicate 95% confidence intervals on 100 repeated and independent experiments. Budget size is on the X-axis. Y-axis in log scale for panels 6e and 6f.

the binary one. Therefore, by training the surrogate on 2500 points we are able to (i) find 90% of true positive calibrations (Figure 6d) and predict the thickness of the associated distribution of growth rates incurring in a mean squared error of less than 0.08 (Figure 6b) using a continuous target variable

Surrogate Algorithm	True Negatives	False Positives	False Negatives	True Positives	Precision
Logit	62	22	61	355	94.17%
XGBoost	178	17	0	305	94.72%
XGBoost (scaled)	193	2	0	305	99.35%

Table 3: Surrogate modelling performance using the learning procedure presented in this paper.

and, (ii) find 80% of the true positives (panel 6c) and correctly classifying around the 80% of them (panel 6a) using a binary target variable.

Given the satisfactory explanatory performance of the surrogate, do we also achieve considerable improvements in the computational time required to perform such exploration exercises? Figures 6e and 6f provides a positive answer. Indeed, the surrogate is 3750 times faster than the fully-fledge Island agent-based model. Moreover, the increase in speed is considerably larger than in the Brock and Hommes model. This confirms our intuition on the increasing usefulness of our surrogate modeling approach when the computational cost of the ABM under study is higher. Such a result is a desirable property for real applications, where the complexity of the underlying ABM could even prevent the exploration of the parameter space.

5.4 Robustness analysis

We now assess the robustness of our training procedure with respect to different surrogate models. More specifically, we compare the XGBoost surrogate employed in the previous analysis with the simpler and more widely used Logit one. Our comparison exercise is performed in a fully stochastic version of the Island agent-based model, where an additional Monte Carlo (MC) is carried out on the seed parameter governing the stochastic terms of the model. As a sneak preview, we can anticipate that our procedure is pretty robust to different surrogate methods.

We focus on a binary outcome setting (the one delivering worse performances) and we employ the milder condition that the average growth rate must be positive and sustained, i.e. $AGR > 0.5\%$. In this way, the results can be compared to those obtained in the original exercise in Fagiolo and Dosi (2003). We set a budget of 500 evaluations of the "true" Islands ABM and run a Monte Carlo exercise of size 100 per parameter combination to generate an MC average of the GDP growth rate that serves as our output variable. Note that this exercise is more complete that the one performed in the previous sections: here, we develop a surrogate model that learns the relationship between parameters and the MC average over their ABM evaluations. This requires many more evaluations of the parameter combinations in the true ABM to converge to the statistic required for the label. In our proposed procedure, an MC average growth rate below 0.5% is labelled "false", while AGR above 0.5% are labeled "true". The aim is to learn a surrogate model that accurately classifies parameter combinations as positive or negative calibrations.

We demonstrate the performance of our machine learning approach using two different surrogates: the nonlinear XGBoost and faster, linear, Logit. The former, employed in the analyses carried out in the previous sections, benefits from increased accuracy in exchange for greater computational costs. The latter is a standard statistical model employed regularly for this type of regression analysis. The performance of these alternative surrogates will be evaluated according to the F1-score while training the surrogate, with the final objective of maximizing the precision of the resulting models, i.e. the number of true evaluations which are accurately predicted as positive before they are evaluated. This is a key point to this exercise because real-world use of the proposed approach does not allow us to

evaluate all the points in our sample space. Real-world evaluation only provides labels for points that are predicted positive and the resulting performance can only be measured with regard to the true and false positives, with a preference to maximize the former.

Using the algorithm described in Section 3, the exercise begins by sampling 1000000 points at random from the Islands parameter space. Two separate instances of the algorithm are compared. One using the nonlinear XGBoost algorithm and another using the standard Logit algorithm. These points are sampled using a standard Sobol sampler. A fixed budget of ABM evaluations is set to 500. This is the maximum number of evaluations we allow before stopping the algorithm. Both algorithms are initialized with 35 labelled parameters, which is the number of random evaluations needed before a single positive calibration was discovered. Remember that the algorithm must be initialised with at least a single positive calibration. Next, a surrogate is fit to these 35 labeled points in this first round and the resulting surrogate is used to predict the label probability for the $1000000 - 35$ remaining unlabeled points. The procedure then samples $log(500)$ combinations at random from the predicted positive combinations. These $log(500)$ combinations are then evaluated in the ABM and added to the set of labeled points, resulting in $35 + log(500)$ labeled points after the first round. This repeats until we achieve 500 labeled combinations from the pool of 1000000 originally unlabelled combinations.

The proposed procedure results in a comparable precision of 94.17% and 94.72% between Logit and XGBoost, respectively. The negligible difference between the precision of the two surrogates suggests that out training procedure provide satisfying results even when the fast and standard Logit statistical model is employed. However, when the XGBoost predicted probabilities are corrected through the Platt scaling procedure,[29] its precision rises to 99.35%. Moreover, scaled XGBoost performs is considerably superior to Logit with regard to true vs. false positives. Considering its higher computation costs and need for hyperparameter optimization in using the more precise XGBoost surrogate, users might prefer the faster Logit surrogate when false positives are cheap. Nevertheless, our proposed surrogate modelling procedure works well in both the Logit and XGBoost cases.

6 Discussion and concluding remarks

In this paper, we have proposed a novel approach to the calibration and parameter space exploration of agent-based models, which combines the use of supervised machine learning and intelligent sampling to construct a cheap surrogate meta-model. To the best of our knowledge, this is the first attempt to exploit machine learning techniques for calibration and exploration in an agent-based framework.

The results obtained with two agent-based models — the Brock and Hommes (1998) asset pricing model and the "Islands" endogenous growth model (Fagiolo and Dosi, 2003) — show that our *machine-learning surrogate* approach provides an accurate proxy of the original model and dramatically reduce the computation time necessary to parameter space exploration and calibration.[30] However, the main

[29] Unlike Logit, which produces accurate probabilities for each of the class labels, probabilities produced by non-parametric algorithms such as XGBoost require scaling. Here, we use Platt Scaling to correct the probabilities produced with XGBoost. Platt-scaling is a way of transforming the outputs of a classification model into a probability distribution over classes. This means that beyond the simple classification, Platt scaling adds a measure of uncertainty over the classification itself. In particular, it works by simply fitting a logit regression to a classifier's score. For more information, see Platt et al. (1999).

[30] In the current work, we also focus on examples dealing with relatively few parameters. This choice is motivated by illustrative reasons and the willingness to use well established models whose code is easily replicable. Further, the results from this paper were produced using a relatively common laptop computer with 16 gigabytes of memory and a 2.4Ghz Intel i7 5500 CPU. The application to a large scale model is currently under development. However, the computational parsimony of the algorithm used to construct our surrogates strongly points to the ability to deal with much richer parameter spaces.

advantage of our methodology remains in its practical usefulness. Indeed, the surrogate can be learnt at virtually zero computational costs (for research applications) and requires a trivial amount of time to predict areas of the parameter space the modeller should focus on with reasonably good results. Furthermore, the usual trade-off between the quantity of information that needs to be processed (computational costs) and the surrogate performance improvements is, in practice, absent. Ultimately, the surrogate prediction exercises proposed in this paper take less than a minute to complete, with the majority of computation coming from the time to assess the budget of true ABM model evaluations. This means, in practical terms, that the modeller can use an arbitrarily large set of parameter combinations and a relatively small training sample to build the surrogate at almost no cost and leverage the resulting meta-model to gain an insight on the dynamics of the parameter space for further exploration using the original ABM.

Relevantly, a rule-of-thumb can be derived from our exercises to determine the budget of evaluations for a given model. To do so, a performance measure (TPR in our case) and a set of parametrizations from the pool to initialize the surrogate can be selected, making sure such set contains at least one positive calibration. Then the procedure described in Section 3.2 should be repeated adding log(pool size) new points at each iteration. The performance v.s. evaluations curve (e.g. Figures 4c and 4d) should be evaluated. Whenever the slope of the curve start to decrease, the user should stop adding new points. The total number of evaluations performed gives, in our view, a reasonable proxy for the budget.

Finally, an additional relevant result emerges from the exercises investigated in this paper. The surrogate is much more effective in reducing the relative cost of exploring the properties of the model over the parameter space for the "Islands" model, which is more computationally intensive than the Brock and Hommes. This suggests that the adoption of surrogate meta-modelling allows to achieve increasing computational gains as the complexity of the underlying model increases.

We believe that our approach offers some advantages over kriging, which has been recently applied to ABMs (Salle and Yildizoglu, 2014; Dosi et al., 2017c, 2016, 2017b; Bargigli et al., 2016). First, our machine learning surrogate works also with large scale agent-based models with more than 30 parameters without introducing additional procedures to select, a priori, the subset of parameters to study, while leaving the rest constant. Second, our approach performs better in out-of-sample testing: the typical Kriging-based meta-model is tested on 10-20 points within an extremely large space, while our surrogate is tested on samples with size 10000 in the first set of exercises and 1000000 points in the last exercise. Finally, as it does not rely on any Gaussianity assumption, the surrogate could provide a better approximation of the rugged, unsmooth surface commonly reported in agent-based models (see e.g. Gilli and Winker, 2003; Fabretti, 2012; Lamperti, 2016). The results contained in Appendix B appear indeed to show that with respect to kriging, our machine learning surrogate exhibits higher precision in predicting the response surface of the Island model and, additionally, it performs the task more efficiently in terms of computational time.

This work is only the first step towards a fully-fledge assessment of the properties of agent-based models employing machine-learning techniques. Such developments are especially important for complex macroeconomic agent-based models (see e.g. Dosi et al., 2010, 2013, 2015, 2017a; Popoyan et al., 2017b) as they could allow the development of a standardized and robust procedure for model calibration and validation, thus closing the existing gap with Dynamics Stochastic General Equilibrium models (see Fagiolo and Roventini, 2017, for a critical comparison of ABM and DSGE models). Consistently, in our future research, we plan to apply our methodology to models of larger scale than

those used in this paper and, additionally, to couple the use of the algorithm of Section 3 with some standard calibration approaches (e.g. Method of Simulated Moments) to construct the calibration measure and the calibration criterion. Further, we are currently extending our approach to perform a complete global sensitivity analysis of agent-based models. Finally, a user-friendly Python surrogate modelling library will also be released for general use.

Acknowledgements

We would like to thank Daniele Giachini, Mattia Guerini, Matteo Sostero, Baláz Kégl, Herbert Dawid and three anonymous referees for their comments. A special thanks goes to Antoine Mandel, who engaged in fruitful discussions and provided valuable insights and suggestions. Further, we would like to thank all the participants in seminars and workshops held at Scuola Superiore Sant'Anna (Pisa), PARIS-SACLAY Center for Data Science (CDS), CNRS, the 2016 CDS Collaborative Hackathon for Macroeconomic Agent-Based Model Surrogates, the 2016 CDS workshop on Macroeconomic Surrogates for Agent-Based Models, the XXI WEHIA conference (Castellon), the XXII CEF conference (Bordeaux), the NIPS 2016 "What If? Inference and Learning of Hypothetical and Counterfactual Interventions in Complex Systems" workshop (Barcelona), the CCS 2016 conference (Amsterdam) and 2016 Paris-Bielefeld Workshop on Agent-Based Modeling (Paris). FL acknowledges financial support from European Union's FP7 project IMPRESSIONS (G.A. No 603416). AS acknowledges financial support from the H2020 project DOLFINS (G.A. No 640772) and H2020 ISIGROWTH (G.A. No 649186), along with hardware support from NVIDIA Corporation, AdapData Paris and the Grid5000 testbed for this research. AR acknowledges financial support from European Union's FP7 IMPRESSIONS, H2020 DOLFINS and H2020 ISIGROWTH projects.

References

Alfarano, S., Lux, T., and Wagner, F. (2005). Estimation of agent-based models: The case of an asymmetric herding model. *Computational Economics*, 26(1):19–49.

Alfarano, S., Lux, T., and Wagner, F. (2006). Estimation of a simple agent-based model of financial markets: An application to australian stock and foreign exchange data. *Physica A: Statistical Mechanics and its Applications*, 370(1):38 – 42.

Amilon, H. (2008). Estimation of an adaptive stock market model with heterogeneous agents. *Journal of Empirical Finance*, 15(2):342 – 362.

An, G. and Wilensky, U. (2009). From artificial life to in silico medicine. In Komosinski, M. and Adamatzky, A., editors, *Artificial Life Models in Software*, pages 183–214. Springer London, London.

Anderson, P. W. et al. (1972). More is different. *Science*, 177(4047):393–396.

Archer, K. J. and Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260.

Assenza, T., Gatti, D. D., and Grazzini, J. (2015). Emergent dynamics of a macroeconomic agent based model with capital and credit. *Journal of Economic Dynamics and Control*, 50:5–28.

Balcan, M.-F., Beygelzimer, A., and Langford, J. (2006). Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72. ACM.

Banerjee, A. V. (1992). A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817.

Barde, S. (2016a). Direct comparison of agent-based models of herding in financial markets. *Journal of Economic Dynamics and Control*, 73:329 – 353.

Barde, S. (2016b). A practical, accurate, information criterion for nth order markov processes. *Computational Economics*, pages 1–44.

Barde, S. and van der Hoog, S. (2017). An empirical validation protocol for large-scale agent-based models.

Bargigli, L., Riccetti, L., Russo, A., and Gallegati, M. (2016). Network Calibration and Metamodeling of a Financial Accelerator Agent Based Model. Working papers, economics, Universitá degli Studi di Firenze, Dipartimento di Scienze per l'Economia e l'Impresa.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bianchi, C., Cirillo, P., Gallegati, M., and Vagliasindi, P. A. (2007). Validating and calibrating agent-based models: a case study. *Computational Economics*, 30(3):245–264.

Booker, A., Dennis, J.E., J., Frank, P., Serafini, D., Torczon, V., and Trosset, M. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13.

Boswijk, H., Hommes, C., and Manzan, S. (2007). Behavioral heterogeneity in stock prices. *Journal of Economic Dynamics and Control*, 31(6):1938 – 1970.

Bottazzi, G. and Secchi, A. (2006). Explaining the distribution of firm growth rates. *The RAND Journal of Economics*, 37(2):235–256.

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

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brock, W. A. and Hommes, C. H. (1997). A rational route to randomness. *Econometrica*, 65(5):1059–1095.

Brock, W. A. and Hommes, C. H. (1998). Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8–9):1235 – 1274.

Brown, D. G., Page, S., Riolo, R., Zellner, M., and Rand, W. (2005). Path dependence and the validation of agent-based spatial models of land use. *International Journal of Geographical Information Science*, 19(2):153–174.

Caiani, A., Godin, A., Caverzasi, E., Gallegati, M., Kinsella, S., and Stiglitz, J. E. (2016a). Agent based-stock flow consistent macroeconomics: Towards a benchmark model. *Journal of Economic Dynamics and Control*, 69:375–408.

Caiani, A., Godin, A., Caverzasi, E., Gallegati, M., Kinsella, S., and Stiglitz, J. E. (2016b). Agent based-stock flow consistent macroeconomics: Towards a benchmark model. *Journal of Economic Dynamics and Control*, 69:375–408.

Carley, K. M., Fridsma, D. B., Casman, E., Yahja, A., Altman, N., Chen, L.-C., Kaminsky, B., and Nave, D. (2006). Biowar: scalable agent-based model of bioattacks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(2):252–265.

Castaldi, C. and Dosi, G. (2009). The patterns of output growth of firms and countries: Scale invariances and scale specificities. *Empirical Economics*, 37(3):475–495.

Chen, S.-H., Chang, C.-L., and Du, Y.-R. (2012). Agent-based economic models and econometrics. *The Knowledge Engineering Review*, 27:187–219.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

Chiarella, C., Iori, G., and Perelló, J. (2009). The impact of heterogeneous trading rules on the limit order book and order flows. *Journal of Economic Dynamics and Control*, 33(3):525–537.

Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335.

Cireşan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, 15(2):201–221.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, 140(3):640–651.

Dawid, H., Gemkow, S., Harting, P., Van der Hoog, S., and Neugart, M. (2014a). Agent-based macroeconomic modeling and policy analysis: the eurace@ unibi model. Technical report, Bielefeld Working Papers in Economics and Management.

Dawid, H., Harting, P., and Neugart, M. (2014b). Economic convergence: Policy implications from a heterogeneous agent model. *Journal of Economic Dynamics and Control*, 44:54–80.

De Marchi, S. (2005). *Computational and mathematical modeling in the social sciences*. Cambridge University Press.

Dosi, G. (1988). Sources, procedures and microeconomic effects of innovation. *Journal of Economic Literature*, 26:126–71.

Dosi, G., Fagiolo, G., Napoletano, M., and Roventini, A. (2013). Income distribution, credit and fiscal policies in an agent-based Keynesian model. *Journal of Economic Dynamics and Control*, 37(8):1598–1625.

Dosi, G., Fagiolo, G., Napoletano, M., Roventini, A., and Treibich, T. (2015). Fiscal and monetary policies in complex evolving economies. *Journal of Economic Dynamics and Control*, 52(C):166–189.

Dosi, G., Fagiolo, G., and Roventini, A. (2010). Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics and Control*, 34(9):1748–1767.

Dosi, G., Pereira, M., Roventini, A., and Virgillito, M. (2017a). When more flexibility yields more fragility: The microfoundations of keynesian aggregate unemployment. *Journal of Economic Dynamics and Control*, forthcoming.

Dosi, G., Pereira, M., Roventini, A., and Virgillito, M. E. (2016). The Effects of Labour Market Reforms upon Unemployment and Income Inequalities: an Agent Based Model. LEM Working Papers Series 2016-27, Scuola Superiore Sant'Anna.

Dosi, G., Pereira, M., Roventini, A., and Virgillito, M. E. (2017b). Causes and consequences of hysteresis: Aggregate demand, productivity and employment. LEM Working Papers Series 2017-07, Scuola Superiore Sant'Anna.

Dosi, G., Pereira, M. C., and Virgillito, M. E. (2017c). On the robustness of the fat-tailed distribution of firm growth rates: a global sensitivity analysis. *Journal of Economic Interaction and Coordination*, pages 1–21.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

Dupouët, O. and Yıldızoğlu, M. (2006). Organizational performance in hierarchies and communities of practice. *Journal of Economic Behavior & Organization*, 61(4):668–690.

Effken, J. A., Carley, K. M., Lee, J.-S., Brewer, B. B., and Verran, J. A. (2012). Simulating nursing unit performance with orgahead: strengths and challenges. *Computers, informatics, nursing: CIN*, 30(11):620.

Fabretti, A. (2012). On the problem of calibrating an agent based model for financial markets. *Journal of Economic Interaction and Coordination*, 8(2):277–293.

Fagiolo, G., Birchenhall, C., and Windrum, P. (2007). Empirical validation in agent-based models: Introduction to the special issue. *Computational Economics*, 30(3):189–194.

Fagiolo, G. and Dosi, G. (2003). Exploitation, exploration and innovation in a model of endogenous growth with locally interacting agents. *Structural Change and Economic Dynamics*, 14(3):237–273.

Fagiolo, G., Guerini, M., Lamperti, F., Moneta, G., Roventini, A., and Sapio, A. (2017). Validation of agent-based models in economics and finance. LEM Working Papers Series 2017/23, Scuola Superiore Sant'Anna.

Fagiolo, G., Napoletano, M., and Roventini, A. (2008). Are output growth-rate distributions fat-tailed? some evidence from oecd countries. *Journal of Applied Econometrics*, 23(5):639–669.

Fagiolo, G. and Roventini, A. (2012). Macroeconomic policy in dsge and agent-based models. *Revue de l'OFCE*, 124:67–116.

Fagiolo, G. and Roventini, A. (2017). Macroeconomic policy in dsge and agent-based models redux: New developments and challenges ahead. *Journal of Artificial Societies and Social Simulation*, 20(1).

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition.

Fernández-Villaverde, J., Rubio-Ramírez, J. F., and Schorfheide, F. (2016). Solution and estimation methods for dsge models. In Taylor, J. and H, U., editors, *Handbook of Macroeconomics*, volume 2, pages 527–724. Elsevier.

Franke, R. (2009). Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, 16(5):804 – 815.

Franke, R. and Westerhoff, F. (2012). Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. *Journal of Economic Dynamics and Control*, 36(8):1193–1211.

Freund, Y. (1990). Boosting a weak learning algorithm by majority. In *COLT*, volume 90, pages 202–216.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156.

Gallegati, M. and Kirman, A. (2012). Reconstructing economics. *Complexity Economics*, 1(1):5–31.

Gilli, M. and Winker, P. (2003). A global optimization heuristic for estimating agent based models. *Computational Statistics & Data Analysis*, 42(3):299 – 312. Computational Ecomometrics.

Goldberg, A. B., Zhu, X., Furger, A., and Xu, J.-M. (2011). Oasis: Online active semi-supervised learning. In *AAAI*.

Grazzini, J. (2012). Analysis of the emergent properties: Stationarity and ergodicity. *Journal of Artificial Societies and Social Simulation*, 15(2):7.

Grazzini, J. and Richiardi, M. (2015). Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control*, 51:148 – 165.

Grazzini, J., Richiardi, M. G., and Tsionas, M. (2017). Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77:26 – 47.

Grimm, V. and Railsback, S. F. (2013). *Individual-based modeling and ecology*. Princeton university press.

Gualdi, S., Tarzia, M., Zamponi, F., and Bouchaud, J.-P. (2015). Tipping points in macroeconomic agent-based models. *Journal of Economic Dynamics and Control*, 50:29 – 61. Crises and ComplexityComplexity Research Initiative for Systemic InstabilitieS (CRISIS) Workshop 2013.

Guerini, M. and Moneta, A. (2016). A Method for Agent-Based Models Validation. LEM Papers Series 2016/16, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy.

Herlands, W., Wilson, A., Nickisch, H., Flaxman, S., Neill, D., Van Panhuis, W., and Xing, E. (2015). Scalable gaussian processes for characterizing multidimensional change surfaces. *arXiv preprint arXiv:1511.04408*.

Ilachinski, A. (1997). Irreducible semi-autonomous adaptive combat (isaac): An artificial-life approach to land warfare. Technical report, DTIC Document.

Issaks, E. H. and Srivsatava, R. M. (1989). *Applied geostatistics*. Oxford University Press.

Kukacka, J. and Barunik, J. (2016). Estimation of financial agent-based models with simulated maximum likelihood. IES Working Paper 7/2016, Charles University of Prague.

Lamperti, F. (2016). Empirical Validation of Simulated Models through the GSL-div: an Illustrative Application. LEM Papers Series 2016/18, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy.

Lamperti, F. (2017). An information theoretic criterion for empirical validation of simulation models. *Econometrics and Statistics*, forthcoming.

Lamperti, F., Dosi, G., Napoletano, M., Roventini, A., and Sapio, A. (2017). Faraway, so close: coupled climate and economic dynamics in an agent based integrated assessment model. LEM Working Papers Series 2017/12, Scuola Superiore Sant'Anna.

Lamperti, F. and Mattei, C. E. (2016). Going Up and Down: Rethinking the Empirics of Growth in the Developing and Newly Industrialized World. LEM Papers Series 2016/01, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy.

Leal, S. J., Napoletano, M., Roventini, A., and Fagiolo, G. (2014). Rock around the clock: an agent-based model of low-and high-frequency trading. *Journal of Evolutionary Economics*, pages 1–28.

Lee, J.-S., Filatova, T., Ligmann-Zielinska, A., Hassani-Mahmooei, B., Stonedahl, F., Lorscheid, I., Voinov, A., Polhill, J. G., Sun, Z., and Parker, D. C. (2015). The complexities of agent-based modeling output analysis. *Journal of Artificial Societies and Social Simulation*, 18(4):4.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013). Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439.

Lux, T. and Marchesi, M. (2000). Volatility clustering in financial markets: a microsimulation of interacting agents. *International journal of theoretical and applied finance*, 3(04):675–702.

Macy, M. W. and Willer, R. (2002). From factors to actors: Computational sociology and agent-based modeling. *Annual review of sociology*, pages 143–166.

Marks, R. E. (2013). Validation and model selection: Three similarity measures compared. *Complexity Economics*, 2(1):41–61.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Morokoff, W. J. and Caflisch, R. E. (1994). Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279.

Moss, S. (2008). Alternative approaches to the empirical validation of agent-based models. *Journal of Artificial Societies and Social Simulation*, 11(1):5.

Mullainathan, S. and Spiess, J. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Petrovic, S., Osborne, M., and Lavrenko, V. (2011). Rt to win! predicting message propagation in twitter. *ICWSM*, 11:586–589.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Popoyan, L., Napoletano, M., and Roventini, A. (2017a). Taming macroeconomic instability: Monetary and macro-prudential policy interactions in an agent-based model. *Journal of Economic Behavior & Organization*, 134:117–140.

Popoyan, L., Napoletano, M., and Roventini, A. (2017b). Taming Macroeconomic Instability: Monetary and Macro Prudential Policy Interactions in an Agent-Based Model. *Journal of Economic Behavior & Organization*, 134:117–140.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.

Recchioni, M. C., Tedeschi, G., and Gallegati, M. (2015). A calibration procedure for analyzing stock price dynamics in an agent-based framework. *Journal of Economic Dynamics and Control*, 60:1 – 25.

Ross, S., Gordon, G. J., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, volume 1(2), page 6.

Roy, N. and McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448.

Salle, I. and Yildizoglu, M. (2014). Efficient Sampling and Meta-Modeling for Computational Economic Models. *Computational Economics*, 44(4):507–536.

Sani, A., Lazaric, A., and Ryabko, D. (2015). The replacement bootstrap for dependent data. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 1194–1198. IEEE.

Settles, B. (2010). Active learning literature survey. Technical Report 55-66, University of Wisconsin, Madison.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM.

Squazzoni, F. (2010). The impact of agent-based models in the social sciences after 15 years of incursions. *History of Economic Ideas*, pages 197–233.

Sridharan, M. and Tesauro, G. (2002). Multi-agent q-learning and regression trees for automated pricing decisions. In *Game Theory and Decision Theory in Agent-Based Systems*, pages 217–234. Springer.

Subbotin, M. T. (1923). On the law of frequency of error. *Matematicheskii Sbornik*, 31(2):296–301.

ten Broeke, G., van Voorn, G., and Ligtenberg, A. (2016). Which sensitivity analysis method should i use for my agent-based model? *Journal of Artificial Societies & Social Simulation*, 19(1).

Tesfatsion, L. and Judd, K. L. (2006). *Handbook of computational economics: agent-based computational economics*, volume 2. Elsevier.

Thiele, J. C., Kurth, W., and Grimm, V. (2014). Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using netlogo and r. *Journal of Artificial Societies and Social Simulation*, 17(3):11.

Timothé, C. and Olivier, P. (2015). Optimism in active learning.

Weeks, M. (1995). Circumventing the curse of dimensionality in applied work using computer intensive methods. *The Economic Journal*, 105(429):520–530.

Wilson, A. G., Dann, C., and Nickisch, H. (2015). Thoughts on massively scalable gaussian processes. *arXiv preprint arXiv:1511.01870*.

Winker, P., Gilli, M., and Jeleskovic, V. (2007). An objective function for simulation based inference on exchange rate data. *Journal of Economic Interaction and Coordination*, 2(2):125–145.

Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. In *Soft Computing and Industry*, pages 25–42. Springer.

Zhu, X. (2005). Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison.

Appendices

A Surrogate Modeling Methodology: Technical Details

Set:

- Evaluation budget B
- Agent Based Model ABM, characterized by the d-dimensional parameter vector $X \in \mathbb{R}^d$
- Surrogate Modeling Algorithm $\mathcal{A}$, characterized by parameters θ[a]

Initialize:

- Draw a large pool of parameter combinations $\bar{\mathbf{X}}$ uniformly at random.
- Draw a small subset $\mathbf{X}$ uniformly at random from $\bar{\mathbf{X}}$.

While $|Y| < B$**, repeat**

1. **For** X **in** $\mathbf{X}$**:**
   - Evaluate $Y = \text{ABM}(X)$
   - Add $\{X, Y\}$ to the set of evaluated samples $\mathbf{Z}$
   - Remove X from $\bar{\mathbf{X}}$
2. Given the evaluated set $\mathbf{Z}$, find optimized surrogate parameters $\hat{\Theta} = \text{HPO}(\mathcal{A}(\Theta, \mathbf{Z}))$[b]
3. Given $\mathcal{A}(\hat{\Theta})$, predict $\hat{\mathbf{Y}}$
4. Select a small subset $\mathbf{X}$ from $\bar{\mathbf{X}}$ conditioned on $\hat{\mathbf{Y}}$

end while

[a] We refer to these as "hyper" parameters because they are parameters of the surrogate model and not the ABM. Therefore they are of indirect concern and change according to the choice of surrogate modeling algorithm.

[b] Here, we automatically set parameters ("Hyper-parameter Optimization") of the machine learning algorithm using the Nelder-Mead downhill simplex method, a standard numerical optimization approach for minimizing or maximizing a high-dimensional function (Nelder and Mead, 1965).

Figure 7: Pseudo-code for our surrogate model training algorithm.

At each update of the surrogate model, the learning objective is defined by minimizing the following objective,

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta),$$

where L and Ω are the selected loss and regularization functions, respectively, over machine learning parameters Θ. Recall that we set L to the F_1 score for a binary response and the MSE for a real-valued response. The selected surrogate algorithm in this paper is the XGBoost algorithm (Chen and Guestrin, 2016), which is an implementation of extreme gradient boosted decision trees. [31] Recall that each node in the tree splits a specific feature according to a a simple $\geq$ or $<$ condition to segregate the parameter vector along the path from the top "root" node until it reaches the final "leaf" node, where the result of the path is a leaf that denotes the predicted response of the parameter vector. An example of a single decision tree can be seen in Figure 8. XGBoost constructs a set of these trees over multiple rounds with the aim of optimizing the above objective function. The complexity of the surrogate model is controlled by the regularization term, which prevents "overfitting" the set of

[31] For more information on Boosting, see Freund (1990); Freund et al. (1996); for more information on CART trees, see Breiman et al. (1984).

in-sample parameter combinations. In the case of XGBoost, Ω is a parameterized mixture between L_1 and L_2 regularizers.

In particular, the surrogate predicts the true response y_i as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F},$$

where K is the number of CART trees in the ensemble and $\mathcal{F}$ is the functional space of all possible CART trees. Given that CART trees are functions, the loss gradient resulting from an ensemble or set of CART trees can be used to optimize any selected target objective.

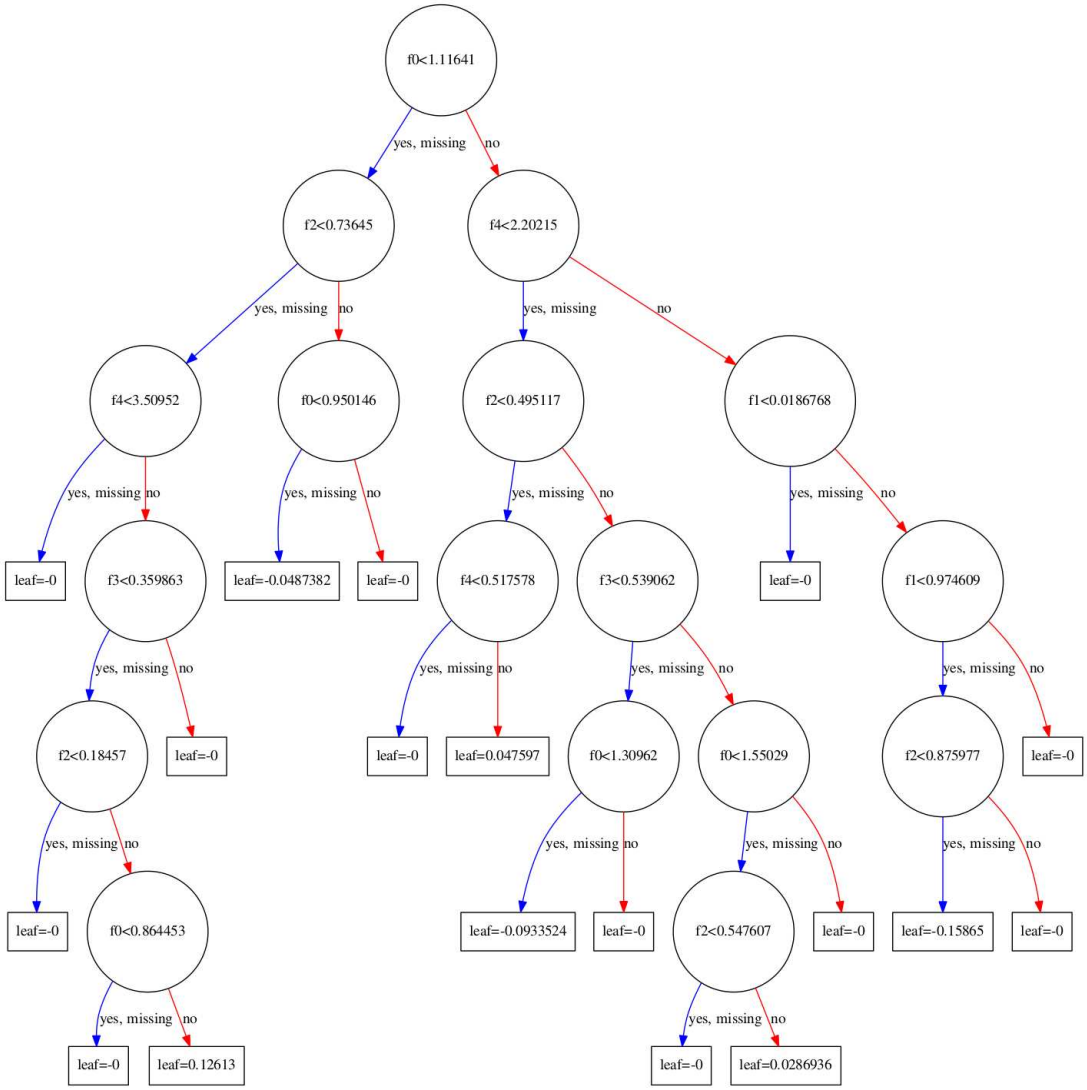


Figure 8: An example classification and regression tree (CART) used for regression. Features are labelled $f0, \ldots, f4$ and nodes specify cutoff thresholds that designate the path a new parameter vector takes from the top (root) node to the final (leaf) node, which denotes the predicted calibration value. In the process of "boosting" CART trees to produce an ensemble, each subsequent tree increasingly focuses on the higher weighted samples. This generally results in smaller "specialized" trees that stick on samples that were most difficult to classify.

In the case of XGBoost, "boosting" is used to magnify the importance of difficult-to-learn parameter combinations. Note that the algorithm only uses evaluated parameter combinations, so the

response is available for all the parameter combinations used to approximate the response. Recall that the algorithm splits the evaluated parameter combinations into an in-sample and out-of-sample set and only approximates the response using the in-sample set. The out-of-sample set is then used to evaluate the performance of the algorithm according to the selected objective. Boosting uses weights to bias hard to approximate parameter combinations in this set of in-sample combinations according to their difficulty over previously learned decision trees in the set. Note that the algorithm iteratively constructs decision trees and each subsequent tree only approximates the performance based on the biased weighting over parameter combinations. This translates to a different loss over the parameter combinations in each of the rounds and tree "boosting" in the direction of the gradient that minimizes the total loss. Accordingly, the CART trees increasingly specialize to handle the subset of parameter combinations that were particularly difficult to approximate (see Freund, 1990; Freund et al., 1996; Chen and Guestrin, 2016, for more details see).

The iterative construction of the ensemble begins from a first CART prediction,

$$\hat{y}_i^{(0)} = 0,$$

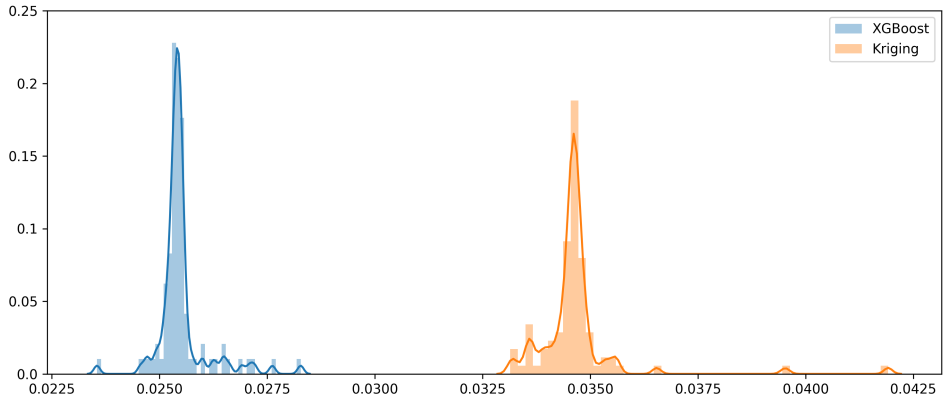
with subsequent trees,

$$\begin{aligned}
\hat{y}_i^{(1)} &= f_1(x_i) \\
&= \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\vdots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) \\
&= \hat{y}_i^{(t-1)} + f_t(x_i).
\end{aligned}$$

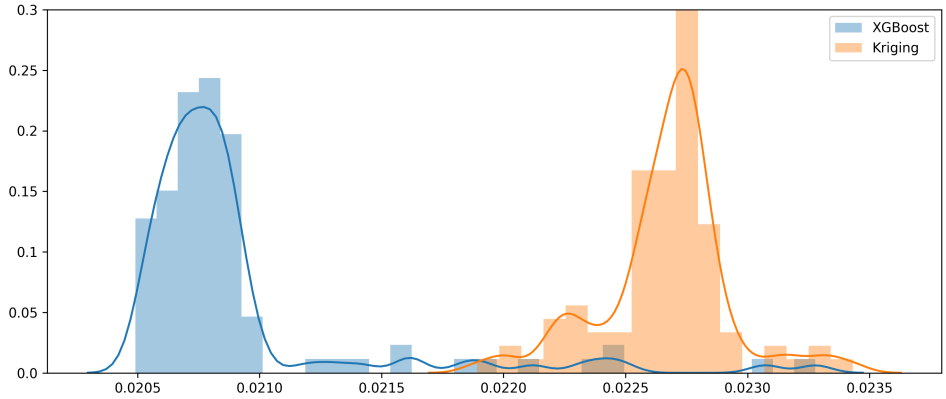
Once CART t is completed, its prediction $\hat{y}_i^{(t)}$, are "boosted" by reweighting the set of evaluated parameter combinations x_i, according to the aggregate prediction performance of the ensemble of trees constructed up to the current round. Given this ensemble at $t-1$, a new CART is added according to its ability to minimize,

$$\begin{aligned}
Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\
&= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant.
\end{aligned}$$

Further, as each tree is added in a sequence of trees that depend on the aggregate performance over previously constructed trees, the nodes of subsequent trees increasingly focus on the difficult samples. The result is a set of trees that grows in predictive power by exploiting the knowledge of errors in previous rounds.



(a) Budget = 50 ABM evaluations



(b) Budget = 500 ABM evaluations

Figure 9: An out-of-sample comparison of the Mean Squared Error (MSE, on the X-axis) performance in the real-valued outcome setting over 100 independently drawn out-of-sample sets of 100 parameter combinations between the proposed approach and Kriging on the Islands Model. The methods are compared using 50 ABM evaluations in the upper plot and 500 ABM evaluations in the lower plot.

B Kriging vs. Surrogate Meta-Modeling Methodology: Budgeted Performance Comparison

Here we report the results of a comparison exercise we ran to test our procedure against kriging in providing a good meta-model for the Islands ABM within the real-valued outcome setting.[32] The plots in Figure 9 compare the distribution of mean-squared errors of kriging and our procedure (labelled as XGBoost) over 100 random out-of-sample sets of 100 parameter combinations selected using Sobol Sampling from the Parameters in Table 2, with the slight adjustment that the number of periods $T_{IS} = 100$. We find that the XGBoost surrogate learned though our approach appear to outperform kriging with regard to the out-of-sample mean squared error. Further, the performance appears to be consistently superior shifting from a small budget of 50 evaluation to a larger one of 500.

Ignoring the computation cost of evaluating the ABM, the run time for our surrogate proposed approach scales on the order of $\mathcal{O}(|X|)$, where X is the total number of parameters.[33] Instead, Kriging

[32] Kriging has been implemented using the Python-based toolbox PyKriging. PyKriging is available at https://github.com/capaulson/pyKriging. The full code to replicate the exercise in this Appendix can be found at https://github.com/amirsani/online_surrogate_modeling.

[33] Chen and Guestrin (2016) show that a single evaluation of the XGBoost algorithm requires $\mathcal{O}(Kd|X|)$, where $K \ll |X|$ is the total number of trees, $d \ll |X|$ is the maximum depth of all the trees and $|X|$ is the total number of data points. Note also that the algorithm is run over $T = \left\lceil \frac{B}{\log(B)} \right\rceil \lll |X|$ iterations, so $|X|$ is still the dominating term.

is expected to scale on the order of $\mathcal{O}(|X|^3)$ for each point in X as the inverse and determinant of the selected kernel needs to be computed along with the derivatives of the log likelihood (Issaks and Srivsatava, 1989; Rasmussen and Williams, 2006). As the number of parameters ($|X|$) increases, and ignoring the performance advantage of using a non-linear surrogate, the cost of adopting a kriging meta-model is expected to become prohibitive when compared to XGBoost.