Research Article

Validation and Calibration of an Agent-Based Model: A Surrogate Approach

Yi Zhang ^{1,2}, Zhe Li,¹ and Yongchao Zhang³

¹School of Economics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

²College of business administration, Northeastern University, Shenyang 110819, China

³School of Mathematics and Statistics, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Correspondence should be addressed to Yi Zhang; 123811595@qq.com

Received 18 June 2019; Revised 21 August 2019; Accepted 11 September 2019; Published 21 January 2020

Academic Editor: Marko Robnik

Copyright © 2020 Yi Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Agent-based modelling has been proved to be extremely useful for learning about real world societies through the analysis of simulations. Recent agent-based models usually contain a large number of parameters that capture the interactions among microheterogeneous subjects and the multistructure of the complex system. However, this can result in the “curse of dimensionality” phenomenon and decrease the robustness of the model’s output. Hence, it is still a great challenge to efficiently calibrate agent-based models to actual data. In this paper, we present a surrogate analysis method for calibration by combining supervised machine-learning and intelligent iterative sampling. Without any prior assumptions regarding the distribution of the parameter space, the proposed method can learn a surrogate model as the approximation of the original system with a relatively small number of training points, which will serve the needs of further sensitivity analysis and parameter calibration research. We take the heterogeneous asset pricing model as an example to evaluate the model’s performance using actual Chinese stock market data. The results demonstrate the good capabilities of the surrogate model at modelling the observed reality, as well as the remarkable reduction of the computational time for validating the agent-based model.

1. Introduction

Agent-based models (ABMs) are favoured by researchers when explaining the emergence of complex systems [1, 2]. The explanatory power of the existing ABMs mainly comes from exploring the market mechanism by describing heterogeneous agents’ behavioural activities and their interactions, which are widely used in economics, demography, and ecology [3–5]. Since an ABM can reveal the dynamics of complex systems using highly flexible, natural, descriptive ways, many scholars regard it as “one of the most important methods of complex scientific methodology” [6], and some even deem it to be “a revolutionary development for social science” [7].

However, ABMs are criticized for their lack of objective verification criteria, which harm the number and persistence of related studies [8–10]. Some researchers doubt that ABMs can obtain any desirable results in subjective settings, claim that its practical applicability is exaggerated, and believe that the modellers are biased and not objective in the modelling process in order to obtain specific results [11, 12].

Due to the complexity of real systems, ABMs usually contain a large number of parameters that need to be calibrated. As the parameter spaces geometrically expand as the number of parameters increase, it results in another challenge in the use of ABMs, which is referred to as the “dimensional disaster” [13]. It has fairly high hardware requirements and computational costs when searching meaningful parameter combinations, since the parameter space of ABMs cannot be exhausted, which is usually computationally prohibitive for researchers.

Whether an ABM is a good approximation of the original system depends on the verification of the results, which is accomplished by testing the consistency of the statistical characteristics of the ABM’s output with respect to real data. In the high-dimensional parameter space, any estimator converges slowly to the true value of the smoothing function, resulting in the local critical point being mistaken for the global maximum or minimum [14]. Therefore, how to effectively find the parameter space of sensitive parameters and calibrate it has become one of the key problems of AB modelling.

The existing ways of dealing with this issue can be mainly divided into three categories: the indirect calibration method, the Werker-Brenner method, and the historical data method. Here, the historical data method is more prevalent due to its excellent fit and easy verification [15]. The historical data method is implemented by dividing the collected data into a modelling set and a verification set to evaluate the model and verify results, respectively. Gilli and Winker [16] present a continuous global optimization heuristic for estimating the ABM of the foreign exchange market. Khashanah and Alsulaiman [17] develop a multisubject meta-model to capture the complexity of stock markets and calibrate the model using a scatter search heuristic approach. Franke and Westerhoff [18] present an improved structural stochastic volatility model for parameter calibration, but it is considered to be a relative simple model that contains only few parameters. Recchioni et al. [19] propose a calibration method that uses a simple gradient-based algorithm and evaluates the performance based on the out-of-sample prediction errors. Similar research can be found in Fievet and Sornette [20] and Amilon [21].

Recently, the surrogate analysis approach has been used increasingly more in the analysis of ABMs [22, 23]. The main idea of this approach is to generate a surrogate model using a certain learning algorithm as the approximation of the original agent-based model. The surrogate model can reduce the dimensionality of the original model parameter vector and greatly simplify the form while maintaining the dynamic characteristics of the original system.

The key to surrogate model analysis depends on the chosen learning algorithm. In previous research, the main approach was the Kriging linear interpolation method. This approach estimates the ABM output for the parameter space using the ABM evaluation of limited samples, and then generates the best unbiased linear prediction factor by investigating the real variation map or real spatial correlation of the data. Under the condition that the data obey a uniform distribution, Kriging interpolation only needs 30 data points to approximate the spatial structure, which makes it a very efficient technique. However, for most of the complex systems, the distribution of the data is unknown. In this regard, the Kriging method relies on expert knowledge of the variogram to estimate the spatial dependence of the points, which demands a fairly large simulation data set and significantly higher computational costs.

In this paper, we present a new approach for ABM validation and calibration based on the surrogate model. By combining machine learning and the intelligent sampling technique, the method can learn the approximate surrogate model of the original ABM model at relatively low time costs. The main advantage of this method is that it can search the ABM parameter space using fewer computing resources and efficiently find the response surface of the model under fewer constraints. In particular, it does not need to make any prior assumptions about the distribution of the parameter space.

2. Surrogate Model

It is crucial to choose an appropriate learning algorithm for surrogate analysis. In this section, we first define the relevant

concepts of the ABM calibration. Then, we will discuss the details of the CatBoost machine learning algorithm and use it in our work. Finally, the complete procedure of generating a surrogate model based on CatBoost is presented. We should point out that our work is an improvement based on the research work of Lamperti et al. [22], which has combined the xgboost algorithm with intelligent sampling method to generate a fast learning surrogate model for ABMS validation. In our research, we use the newly developed machine learning technique for generating the surrogate model and look forward to obtain some new findings.

2.1. Related Concepts. Whether the ABM outputs are consistent with the real data depends on the “calibration measure”. The ABM outputs can be divided into two types: binary outputs and real value outputs. In the binary case, the calibration measure can take only two values: 1 and 0. A value of 1 means that the statistical characteristics of the output are consistent with the real data, and it is 0 otherwise. For instance, if we test whether the output data that are generated by the ABM have the same fat-tailed characteristic as the real data, and if it does, the calibration measurement takes the value of 1; otherwise it is 0. In the real case, the statistical characteristics of the ABM output data are quantitatively calculated, and the calculation result is used as the calibration measurement. For example, we can assess the kurtosis or the tail index of the output data. We expect to find the parameter vectors whose calibration measurements meet certain specific conditions, and these conditions are called “calibration criteria”. For example, if the modeller wants to test whether the output data have a nonnormal distribution with negative skewness and leptokurtosis, he can use both the negative skewness and leptokurtosis as calibration criteria. For the real output case, the minimization of the loss function can be used as the calibration criterion.

A parameter vector $x \in I$ with a response that accords with the calibration criteria is characterized as “positive calibration” and labelled as positive; otherwise, when the opposite occurs, it receives a negative label. We expect to find the maximum number of positive labelled points in the parameter space and use them in learning to generate the surrogate model.

It should be noted that positive calibrated parameter vectors may be located in multiple discontinuous regions over the entire parameter space rather than a smooth topology. The approach we proposed avoids making any prior assumptions on the response surface of the output, which makes it universal for most of the real applications.

2.2. CatBoost. CatBoost is a supervised machine learning algorithm that executes a process called boosting to classify categorical data. CatBoost is at its core a decision tree boosting algorithm. Boosting refers to the integrated learning method that sequentially establishes a large number of models. In classical GBDT, it is based on using the same data set to obtain the gradient value of the loss function for the current model in each iteration. However, this will cause the model to suffer from over-fitting due to pointwise gradient estimation bias. CatBoost uses the ordered boosting method to modify the gradient estimation in the classical algorithm, and it then obtains the unbiased estimation of the gradient to reduce the influence of the

Input: $(x_i, y_i)_{i=1}^n, M, L(y, F(x))$

- 1: Initialize f_0
- 2: **for** $m = 1 \leftarrow 1$ to M **do**
- 3: $r_{im} = -[\partial L(y_i, F(x_i))/\partial F(x_i)]_{F(x)=F_{m-1}(x)}$ **for** $i = 1, \dots, n$
- 4: $h_m^* = \underset{h}{\operatorname{argmin}} \sum_{i=1}^n L(r_{im}, h_m(x_i))$
- 5: $\alpha^* = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \alpha h_m^*(x_i))$
- 6: $F_m = F_{m-1}(x) + \alpha^* h_m^*(x)$
- 7: **end for**
- 8: **return** $F_M(x)$

ALGORITHM 1: GBDT

Input: Training set $(X_k, Y_k)_{k=1}^n$, Number of iterations I

- 1: $M_i \leftarrow 0$ **for** $i = 1, \dots, n$
- 2: **for** $iter \leftarrow 1$ to I **do**
- 3: **for** $i \leftarrow 1$ to n **do**
- 4: **for** $j \leftarrow 1$ to $i - 1$ **do**
- 5: $g_j \leftarrow (d/da) \text{Loss}(y_i, a)|_{a=M_i(X_j)}$
- 6: **end for**
- 7: $M \leftarrow \text{Learn Model}((X_j, g_j) \text{ for } j = 1, \dots, i - 1)$
- 8: $M_i \leftarrow M_i + M$
- 9: **end for**
- 10: **end for**
- 11: **return** $M_1, \dots, M_n; M_1(X_1), M_2(X_2), \dots, M_n(X_n)$

ALGORITHM 2: Ordered Boosting

gradient estimation bias. In this way, the generalization ability of the model is improved. The algorithm flows of GBDT and ordered boosting are illustrated as shown in Algorithms 1 and 2.

Here, F^i is the model that is built using the first i trees, and $g^i(X_k, Y_k)$ is the gradient value of the k -th training sample. To obtain the unbiased estimation of the gradient value $g^i(X_k, Y_k)$ with respect to the model F^i , the training of F^i should not contain observation X_k . If we extend it to the entire process, no points should be used to train F^i . To deal with this seemingly unsolvable problem, we consider the following trick. We train a separate model M_k for each observation X_k without using any X_k -containing samples, and all M_k s have identical tree structures. Then, we calculate the gradient for X_k and score the resulting tree. We use the F_1 score and root mean square error MSE as the loss functions $\text{Loss}(y, a)$ for the binary and real-value cases, respectively. More details will be discussed later.

2.3. Step-by-Step Implementation. The procedure that we designed for generating the surrogate model is illustrated in Figure 1. Three initial settings should be determined before running the program.

(1) *Select the Surrogate Algorithm.* The modeller must select a learning algorithm to perform as the surrogate for the original agent-based model. We choose CatBoost as our surrogate algorithm for two reasons: it has a remarkably high

computational efficiency, and it does not require too many assumptions about the parameters space.

(2) *Select the Fast Sampling Procedure.* The modeller must determine the sampling method to draw samples from the parameter space for the training of the surrogate model.

(3) *Select the Performance Measure of the Surrogate Model.* The modeller must give a quantitative measurement for assessing the performance of the surrogate.

The surrogate program is implemented step-by-step as follows.

Step 1. Construct a relatively large pool of parameter combinations as a substitute set for the parameter space using a certain sampling routine. To ensure that the parameter pool covers all possible regions of the parameter space without knowing the topology, we use the quasi-random sobol sampling, which is designed to fill the sample space, even at small quantities, and is efficient to implement.

Step 2. Randomly draw a small subset from the parameter pool and run the AB model. Each parameter vector is identified as positive or negative according to the calibration measurement and calibration criterion.

Step 3. The surrogate model is generated by using the labelled points to learn with the CatBoost algorithm. This model is the integration of simple decision trees, which can provide better prediction performance compared to other learning algorithms.

Step 4. Predict and label all of the parameter combinations in the pool according to the results of the surrogate model.

Step 5. Draw a small subset of unlabelled points in step 2 and run the ABM. The points are labelled and added to the training set to construct a new subset of training samples. The newly added parameter vectors are randomly selected from the positive labelled parameter combinations that are predicted by the surrogate model. In this way, the algorithm will gradually increase the included “true” positive labels and exclude the “false” positive labels. If there is no positive point in the present round, an uncertain sampling method is used to add new data points to the parameter pool. The uncertain sampling method increases the sampling frequency of the parameter space that the surrogate model has difficulty correctly predicting based on the entropy of the existing label distribution.

Repeat Steps 3–5 until the budget is reached.

The proposed method can intelligently pick the meaningful parameter combination points in multiple rounds of sampling process, which continuously improves the sampling performance and the calibration accuracy at relatively low computational costs. Compared to other iterative Monte Carlo sampling methods, the advantages of our approach are mainly as follows. First, it does not need to make any assumptions about the parameter distribution. Second, it does not require a prior assumption regarding the approximate distribution of the model’s response. Third, the approach does not require that the points satisfy the Markov chain distribution.

FIGURE 1: The surrogate model generation procedure.

2.4. Model Evaluation. We have to evaluate the surrogate model's performance once it is generated. In the case of a binary

output, we use the F_1 score that was introduced by Fawcett as the calibration measurement [24], which is calculated as follows:

$$F_1 = \frac{2 \times \text{"true" positive labels}}{2 \times \text{"true" positive labels} + \text{"false" positive labels} + \text{"false" negative labels}}, \quad (1)$$

where $F_1 \in (0, 1)$. The larger that F_1 is, the better the surrogate model works. When moved to the real-value setting, we use the mean square error as the loss function:

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

where $\hat{y}_i$ is the predicted value of the surrogate, and N is the number of data points in the learning set, which is the number of parameters that are evaluated by the agent model. Thus, the pursuit of an optimal surrogate model is equal to minimizing the MSE.

Finally, we use the "True Positive Rate", which is calculated using an out-of-sample data set, to measure the surrogate's capacity to find positive labels for both settings (Fawcett, 2006). It is calculated as follows:

$$\text{TPR} = \frac{\text{Number of correctly predicted positives}}{\text{Number of positives in the pool}}. \quad (3)$$

The proposed method also provides us with an instinctive way to assess the importance of parameters on the output by calculating the number of splits of a parameter in the CART tree generation process. Since each tree is built based on the optimal segmentation of the probable values of the parameter

TABLE 1: Model parameters and range settings.

Parameter	Description	Range	Explored range
β	Intensity of the choice	$[0; +\infty)$	$[0.0; 10.0]$
n_1	Proportion of type 1 traders	$[0; 1]$	0.5
b_1	Intercept term of type 1 agents	$(-\infty; +\infty)$	$[-1.5; 1.5]$
b_2	Intercept term of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
a_1	Trend of type 1 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
a_2	Trend of type 2 traders	$(-\infty; +\infty)$	$[-2.0; 2.0]$
C	Transaction costs	$[0; +\infty]$	$[0.0; 0.5]$
ω	Impact weight for past prices	$[0.0; 1.0]$	$[0.0; 1.0]$
σ	Volatility	$[0; +\infty)$	$[0.0; 1.0]$
ν	Risk aversion coefficient	$[0; +\infty)$	$[0; 100]$
r	Risk free return	$(1; +\infty)$	$[1.01; 1.1]$
T_{BH}	Number of trading periods	N	500

vectors and it pays increasingly more attention to the difficult-to-forecast samples, we can rank the model's parameters in terms of their importance and sensitivity to the output by counting the number of splits.

3. Application

The heterogeneous agent models (HAMs) that were employed by Brock and Hommes [25] model the asset pricing mechanism by assessing the interaction among agents with heterogeneous beliefs and strategies. HAMs are powerful at duplicating the stylized facts of financial data series, such as volatility clustering, fat tails, long memory, and the leverage effect. The model is also useful for explaining financial market abnormalities such as bubbles and crashes. Recent evidence proves that HAMs provide empirical results that outperform conventional capital asset pricing models or arbitrage models, which makes this theory one of the representative theories of behavioural finance. We choose heterogeneous agent models as an ideal investigatory instance for two reasons: they have been widely studied by financial researchers and they offer a proper number of parameters [26–28]. This section will first briefly describe the heterogeneous trader pricing model, then use the model to test the performance of our method, and finally report the evaluation and comparison results.

3.1. The Heterogeneous Agent Models. Consider that there are N agents who are engaged in trading activities in a market consisting of risky assets and risk-free assets. We denote p as the price of the risky asset and d as the uncertain dividend. The wealth of the agent at time $t + 1$ is expressed as

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

FIGURE 2: Return rate histogram of real data.

where z_t is the amount of the risky asset that was bought by the trader at time t , and $R = (1 + r)$ is the gross return of the risk-free asset.

Suppose that all the traders are rational traders that seek to maximize their mean-variance using heterogeneous expectations and trading strategies. The expected price and variance of the risky asset at time t is denoted as E_t and V_t , respectively. Then, the optimal demand of $z_{h,t}$ for type h traders is equivalent to solving the following problem:

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

This implies that

$$z_{h,t} = \frac{E_{h,t}(p_{t+1} + d_{t+1} - Rp_t)}{(\nu\sigma^2)}, \quad (6)$$

where ν measures the risk aversion of traders, and σ indicates the conditional volatility, which includes all types of traders and remains constant over time. Under the condition that the quantity supplied of risk assets and the type of traders remain unchanged, the market equilibrium state is calculated as follows:

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

where $n_{h,t}$ denotes the risk asset position that is held by type h agents at time t . If all of the agents are all homogenous traders with rational expectations and the market contains complete information, the no-arbitrage market equilibrium condition can be written as follows:

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

where the expectation is determined by all historical prices and dividends up to time t . We call p^* the fundamental price of the asset. Equation (12) has a unique solution when dividend payments are independently distributed with a constant mean. In this case, the fundamental price is equal to $p^* = E(d_t)/(R - 1)$. The deviation of the real price from the fundamental price can be expressed as $\chi_t = p_t - p_t^*$.

TABLE 2: Descriptive statistics.

	Mean	Median	Maximum	Minimum	Std. dev	Skewness	Kurtosis	P-AC	P-ARCH
P	0.0031	-0.0011	0.0682	-0.0483	0.6222	0.2472	2.1540	0.0421	0.8148
P^*	0.0028	0.0002	0.0632	-0.0412	0.5984	0.2354	2.1354	0.0354	0.7894

FIGURE 3: Parameter importance ranking.

Since different types of traders have heterogeneous expectations regarding stock prices and dividend payments, we can express the gain of the type h traders at time $t + 1$ as follows:

$$E_{h,t}(p_{t+1} + d_{t+1}) = E_t(p_{t+1}^*) + f_h(x_{t-1}, \dots, x_{t-L}), \quad (9)$$

where L is the lag operator, and $f_h(\cdot)$ is a function that represents traders' predictions of future prices. The simple linear expression of function $f_h(\cdot)$ that was proposed by Brock and Hommes (1998) is the following:

$$f_{h,t} = a_h x_{t-1} + b_h, \quad (10)$$

where a_h and b_h are the trend coefficient and intercept term, respectively. The agent is defined as a positive feedback trader if $a_h > 0$, and otherwise they are a negative feedback trader. When $a_h = b_h = 0$, the trader adopts a fundamental trading strategy that believes that the price will converge on the fundamental value.

Following existing studies, we consider that a typical market consists of a fundamentalist and chartist, and $f_{1,t}$ and $f_{2,t}$ are their respective trend functions. The market price can be written as follows:

$$Rx_t = n_{1,t} f_{1,t} + n_{2,t} f_{2,t}. \quad (11)$$

To maximize profits, the traders choose and shift between the two strategies, which is equal to maximizing the following objective function:

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

where C_h is the transaction costs, and $\omega \in [0, 1]$ is the impact weight of past profits. The probability of choosing strategy h for a trader is given by the following:

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

Equation (17) is also known as the market fraction model, where $\beta \in [0, +\infty)$ is the intensity of choice. A larger β suggests more frequent shifting between the two strategies. In this way, the model captures the trader's bounded rationality and the effect of their behaviour on the price.

3.2. Model Setting. The model has 12 total parameters that need to be estimated, as shown in Table 1. We set the parameter space within the range that is shown in Table 1 according to the existing related work. It can be further expanded or reduced based on the modeller's needs.

We choose the daily data of the Chinese Shanghai and Shenzhen 300 Indexes as the real sample data for the calibration. The sample interval is from Jan. 4th, 2017 to Dec. 31th, 2018 and it contains 412 total observations, as shown in Figure 2. The statistical characteristics of the sample are reported in Table 2.

It can be seen from Figure 2 that the sample data have a high peak, are fat tailed, and are right-skewed. Table 2 confirms these phenomena from Figure 2, and the data series exhibits a significant ARCH effect.

FIGURE 4: Surrogate analysis results of the HAM model. (a) F1 score of the binary case. (b) MSE of the real-value case. (c) TPR of the binary case. (d) TPR of the real-value case. (e) Time costs of the binary case. (f) Time costs of the real-value case.

In the binary case, we use the two-sample Kolmogorov-Smirnov method to test whether the distribution of the model's output is consistent with the real data as follows:

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

where r denotes the log return, and F_{RW} and F_S are the distribution functions of the real sample data and simulation data, respectively.

To provide a direct comparison, we use the p value of the Kolmogorov-Smirnov test statistic $p(D > D_{RW,S})$ as the calibration criterion when we analyse the real-value case. The higher the p -value is, the better the fitting effect.

The surrogate model is trained 500 times using different numbers of parameter combinations ranging from 250 to 2500 with 250 samples added in each round. A well-distributed out-of-sample data series is necessary and crucial for evaluating the performance of the model. We set a relatively large number of 100000 unlabelled parameter combinations as the evaluation set of the model, which is based on recent literatures.

4. Results

The importance of the parameters is evaluated and ranked according to the number of splits of each parameter in the decision tree construction process, as shown in Figure 3. The results indicate that the trend coefficients a_1 and a_2 have the most significant impacts on the output, and second is the intensity of choice term β . The intercept terms b_1 and b_2 also have certain impacts on the fit of the model. The risk aversion coefficient ν , the conditional volatility σ , and the wealth regression coefficient ω are relatively less important on the output.

The surrogate model is generated using the procedure that is described in section 3 and the simulation results are shown in Figure 4.

In the case of a binary output, the F_1 score increases as the amount training sample data increases. The F_1 score reaches its maximum at approximately 0.8 when 2500 training samples are used, and the TPR index is approximately 0.75. Since the TPR cannot be greater than 1, we consider the results satisfying.

The surrogate model provides superior results in the real-value setting. Even when the number of training samples is low (500), a higher TPR (approximately 70%) can be obtained. When 2,500 training points are employed, it can reach 95%. This can be explained as the learning process over the continuous variable containing more information about the original system, which leads to better performance compared to the binary case.

Finally, we compare the time costs by running the procedure 100 times and taking the average (in seconds) for each subroutine. The subroutine includes training the surrogate model, predicting the parameter using the surrogate model and labelling the parameter using the ABM. The results show that the time costs of the surrogate method are about one-five-hundredth of those of the original ABM model, which is a remarkable efficiency improvement for parameter calibration.

It should be pointed out that the training sample size in the surrogate model is crucial to the performance of our

model. However, the determination of the sample size still lacks an objective basis and it mainly relies on the rule of thumb. To do this, we should ensure that at least one parameter combination that satisfies the positive calibration criterion is contained in the pool, and, then, a small number of parameter points can be continuously added to the pool during each computing round. When the TPR curve tends to be flat or even tends to decline, the relative size of the training sample can be regarded as a reference for the settings.

5. Conclusion

The agent-based model has been extensively utilized in complex system such as those in economics, demography and management science due to its advantageous high degree of flexibility and freedom. However, there is still a lack of effective parameter calibration methods due to computational restrictions. This paper proposes a surrogate model approach for exploring and calibrating ABM parameters by combining supervised-machine learning with intelligent sampling. By using the CatBoost machine learning algorithm, a surrogate model of the original ABM is learned, which allows the modeller to explore and locate the regions that have significant impacts on the output for the parameter space. Generating the surrogate model only requires a small training sample, which can significantly reduce the computational costs compared to other similar approaches. The results that are obtained from the application of the heterogeneous asset pricing model suggest that our approach possesses good performance with respect to both accuracy and costs. Another advantage of our approach is that it does not require any prior assumptions about the distribution of the parameters or the topology of the output space, which makes it more applicable to a wider range of applications.

The approach that we proposed is a powerful tool for addressing the “dimensional disaster” problem that is caused by the parametric explosion in agent-based model. In future research, we plan to use it in more complex systems with more numerous parameter combinations. We also plan to establish an ABM toolbox that contains surrogate modelling, a calibration measure and the calibration criterion for general use in the future.

Data Availability

The data that are used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was financially supported by the China Postdoctoral Science Foundation (Grant no. 2017M621042); Fundamental Research Funds for the Central Universities (Grant no. N162304015; N162304005).

References

- [1] Z. Wang, C. T. Bauch, S. Bhattacharyya et al., “Statistical physics of vaccination,” *Physics Reports*, vol. 664, no. 9, pp. 1–113, 2016.
- [2] M. Perc, J. J. Jordan, D. G. Rand, Z. Wang, S. Boccaletti, and A. Szolnoki, “Statistical physics of human cooperation,” *Physics Reports*, vol. 687, no. 8, pp. 1–51, 2017.
- [3] A. Samitas, S. Polyzos, and C. Siriopoulos, “Brexit and financial stability: an agent-based simulation,” *Economic Modelling*, vol. 69, no. 3, pp. 181–192, 2018.
- [4] M. Cardinot, C. O’Riordan, J. Griffith, and M. Perc, “Evoplex: a platform for agent-based modeling on networks,” *SoftwareX*, vol. 9, no. 1, pp. 199–204, 2019.
- [5] M. Perc, “Stability of subsystem solutions in agent-based models,” *European Journal of Physics*, vol. 39, no. 1, pp. 1–13, 2017.
- [6] C. R. Shalizi, “Methods and techniques of complex systems science: an overview,” in *Complex Systems Science in Biomedicine*, pp. 33–114, Springer, Boston, MA, USA, 2006.
- [7] S. C. Bankes, “Agent-based modeling: a revolution,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 7199–7200, 2002.
- [8] A. Caiani, A. Godin, E. Caverzasi, M. Gallegati, S. Kinsella, and J. E. Stiglitz, “Agent based-stock flow consistent macroeconomics: towards a benchmark model,” *Journal of Economic Dynamics and Control*, vol. 82, no. 11, pp. 375–408, 2016.
- [9] T. Assenza, D. D. Gatti, and J. Grazzini, “Emergent dynamics of a macroeconomic agent based model with capital and credit,” *Journal of Economic Dynamics and Control*, vol. 50, no. 3, pp. 5–28, 2015.
- [10] L. Popoyan, M. Napoletano, and A. Roventini, “Taming macroeconomic instability: monetary and macro-prudential policy interactions in an agent-based model,” *Journal of Economic Behavior & Organization*, vol. 134, no. 2, pp. 117–140, 2017.
- [11] S. J. Leal, M. Napoletano, A. Roventini, and G. Fagiolo, “Rock around the clock: an agent-based model of low-and high-frequency trading,” *Journal of Evolutionary Economics*, vol. 26, no. 1, pp. 49–76, 2016.
- [12] F. Lamperti, A. Roventini, and A. Sani, “Agent-based model calibration using machine learning surrogates,” *Journal of Economic Dynamics and Control*, vol. 90, no. 10, pp. 366–389, 2018.
- [13] J. S. Lee, T. Filatova, A. L. Zielinska et al., “The complexities of agent-based modeling output analysis,” *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 4, pp. 1–26, 2015.
- [14] J. Grazzini, M. G. Richiardi, and M. Tsionas, “Bayesian estimation of agent-based models,” *Journal of Economic Dynamics and Control*, vol. 77, no. 4, pp. 26–47, 2017.
- [15] G. Fagiolo, C. Birchenhall, and P. Windrum, “Empirical validation in agent-based models: introduction to the special issue,” *Computational Economics*, vol. 30, no. 3, pp. 189–194, 2007.
- [16] M. Gilli and P. Winker, “A global optimization heuristic for estimating agent based models,” *Computational Statistics & Data Analysis*, vol. 42, no. 3, pp. 299–312, 2003.
- [17] K. Khashanah and T. Alsulaiman, “Network theory and behavioral finance in a heterogeneous market environment,” *Complexity*, vol. 21, no. S2, pp. 530–554, 2016.
- [18] R. Franke and F. Westerhoff, “Structural stochastic volatility in asset pricing dynamics: estimation and model contest,” *Journal of Economic Dynamics and Control*, vol. 36, no. 8, pp. 1193–1211, 2012.
- [19] M. C. Recchioni, G. Tedeschi, and M. Gallegati, “A calibration procedure for analyzing stock price dynamics in an agent-based framework,” *Journal of Economic Dynamics and Control*, vol. 60, no. 11, pp. 1–25, 2015.
- [20] L. Fievet and D. Sornette, “Calibrating emergent phenomena in stock markets with agent based models,” *PLoS One*, vol. 13, no. 3, Article ID e0193290, 2018.
- [21] H. Amilon, “Estimation of an adaptive stock market model with heterogeneous agents,” *Journal of Empirical Finance*, vol. 15, no. 2, pp. 342–362, 2008.
- [22] F. Lamperti, A. Roventini, and A. Sani, “Agent-based model calibration using machine learning surrogates,” *Journal of Economic Dynamics & Control*, vol. 90, no. 3, pp. 366–389, 2018.
- [23] S. V. D. Hoog, “Surrogate modelling in (and of) agent-based models: a prospectus,” *Computational Economics*, vol. 53, no. 3, pp. 1245–1263, 2019.
- [24] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [25] W. A. Brock and C. H. Hommes, “Heterogeneous beliefs and routes to chaos in a simple asset pricing model,” *Journal of Economic Dynamic and Control*, vol. 22, no. 8-9, pp. 1235–1274, 1998.
- [26] J. Arifovic and J. Duffy, “Heterogeneous agent modeling: experimental evidence,” *Handbook of Computational Economics*, vol. 4, no. 9, pp. 491–540, 2018.
- [27] A. H. Alizadeh, H. Thanopoulou, and T. L. Yip, “Investors’ behavior and dynamics of ship prices: a heterogeneous agent model,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, no. 7, pp. 98–114, 2017.
- [28] T. Boppart, P. Krusell, and K. Mitman, “Exploiting MIT shocks in heterogeneous-agent economies: the impulse response as a numerical derivative,” *Journal of Economic Dynamics and Control*, vol. 89, no. 4, pp. 68–92, 2018.