2018 IEEE International Conference on Systems, Man, and Cybernetics

Data-Driven Automatic Calibration for Validation of Agent-Based Social Simulations

Il-Chul Moon
Dept. of Industrial and Systems Eng.
KAIST
Daejeon, Republic of Korea
icmoon@kaist.ac.kr

Dongjun Kim
Dept. of Industrial and Systems Eng.
KAIST
Daejeon, Republic of Korea
dongjoun57@kaist.ac.kr

Tae-Sub Yun
Dept. of Industrial and Systems Eng.
KAIST
Daejeon, Republic of Korea
andy2402@kaist.ac.kr

Jang Won Bae
Smart Data Research Group
ETRI
Daejeon, Republic of Korea
jwbae@etri.re.kr

Dong-oh Kang
Smart Data Research Group
ETRI
Daejeon, Republic of Korea
dongoh@etri.re.kr

Euihyun Paik
Smart Data Research Group
ETRI
Daejeon, Republic of Korea
ehpaik@etri.re.kr

***Abstract*—Though agent based models are used in many domains, the usage have been either very abstract model for conceptual experiments or very detailed models with huge engineering efforts in their modeling and calibration. One reason of this limited usage comes from the difficulties in calibrating and validating the model with observed data because the models are very generative in its nature with many hand-picked parameters. This paper presents a noble framework of augmenting machine learning techniques to agent-based models for better calibration and validation. The framework identifies periods of deviation between the simulation and the observation with hierarchical Dirichlet process hidden Markov Model, or HDP-HMM, and the framework automatically calibrates the temporal macro parameters by searching parameter spaces with more likelihoods of validation. After iterations of this framework, our experiments demonstrated sucessful validations on a hypothetical simple segregation model as well as a real world real estate model. This framework is generally usable in any agent based models with temporal macro parameters, which could be true in many existing models.**

***Index Terms*—Agent Based Model, Simulation Validation, Parameter Calibration, Social Simulation**

I. Introduction

Social simulations have been developed and utilized in the field of traffic, military, economics, management, etc [1], but the validation on the social simulations have always been a challenge [2]. In contrast to the simulations in the engineering field, the social simulations have to include the models on human behaviors, and this human behavior modeling can be a source of such validation errors [3][4]. Additionally, the completeness of the simulation model becomes another error source of the validation on social simulation because it is difficult to set the modeling scope of the social simulation environment compared to the controlled environment of the engineering domain. Given these various sources of errors, some modeling experts might argue that the social simulations are designed to generate hypotheses to experiment and analyze further with different methodologies [5]. This perspective does not claim the simulation realism in the quantitative accuracy, and the perspective only emphasizes the response trend of the social simulation models. While we acknowledge this perspective to be one way of appreciating the social simulations, still the direct validation between simulation results and real world measurements would expand the utility of the social simulations.

If we pursue the direct validation at the numerical measure level, not the trend comparison, we should resolve the potential errors of human behavior modeling and noisy social environment. Previous approaches emphasizes the model-based approach by expanding and correcting the behavior model and the environment model until the validation is satisfied [6]. However, such model expansion is not always feasible because of its model development cost and potential undesired effect when the model is altered. Also, the validation can be improved by the carefully calibrating the model parameters, but the parameter search space is too large, particularly when we set the dynamic parameter over the simulation period.

Given the limitations from the model-based approach and the manual parameter calibration, this paper proposes a data-driven automatic calibration on the dynamic parameters of the agent based social simulations. Specifically, we select and calibrate a set of model parameters dynamically to improve the validation result over the simulation period. Typically, the model parameters are provided to the simulation at the initial time, and there was no intervention in the middle of the simulation period. We experimented the dynamic calibration of such parametes, which are used to be static. In the process of the dynamic calibration, we utilized a combination of machine learning algorithms, and we devised an algorithm to merge the learning models and the simulation models in order to increase the validation performance.

To confirm the improved validation, we tested our proposed algorithm merging the simulation model and the learning model with two case studies. Firstly, we experimented the

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00278

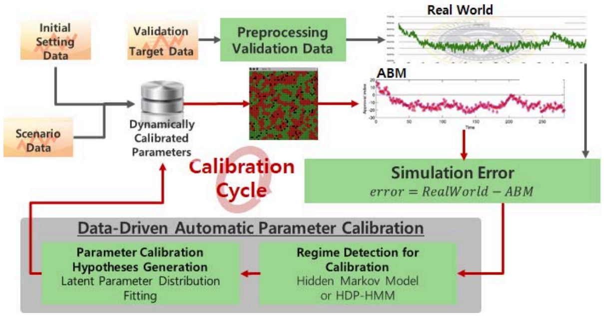

Fig. 1. Overview of Automatic Parameter Calibration

calibration performance with a hypothetical and well-known simulation model, Schelling's segregation model. In the hypothetical experiment, we arbitrarily varied a static pararmeter over the course of simulations. Then, our calibration method is used to detect the varied parameter by only observing the simulation result, and this calibration should result in a better validation against the experimented simulation result. Secondly, we experimented the calibration with a more realistic agent based model in the field of real estates. In this validation to the real world, we cannot anticipate the latent true parameter of the simulation, but we were able to increase the validation level through the dynamic parameter calibration.

II. Previous Researches

Validating a simulation with a real-world dataset has been a holy-grail of the simulation field because a ultimate goal of simulation could be a realistic representation of the target domain.

Given the various domains of simulations, the traffic simulations are one of the most well-validated models with many previous researches. For instance, there are multiple suggestions on the practical calibration procedures to validate simulations in the traffic domains [7][8][9]. Most suggestions calibrate the key input variables, such as origin-destination matrix, and human behavior parameters, but the calibration procedures are tailored to a specific scenario or a simulator, which are not a general approach for any simulations.

When we look for a general approach for parameter calibration in the simulation field, there were a number of researches utilizing a optimization heuristic to increase the validation level. For instance, Nannen and Eiebn calibrated parameters with a genetic algorithm for an agent-based simulation in the economics field. [10]. However, such approaches might require a better guidance in exploring the parameter search space, and we want to develop the method by adopting a machine-learning approach. Furthermore, the previous work only calibrated the initial parameter, and it is very difficult to apply the method if the parameter changes over time.

III. Methodology

This section proposes an algorithm to dynamically calibrate simulation paramters. First, we describe the overview of the algorithm. Then, we introduce the regime detection and the parameter sampling phases in turn.

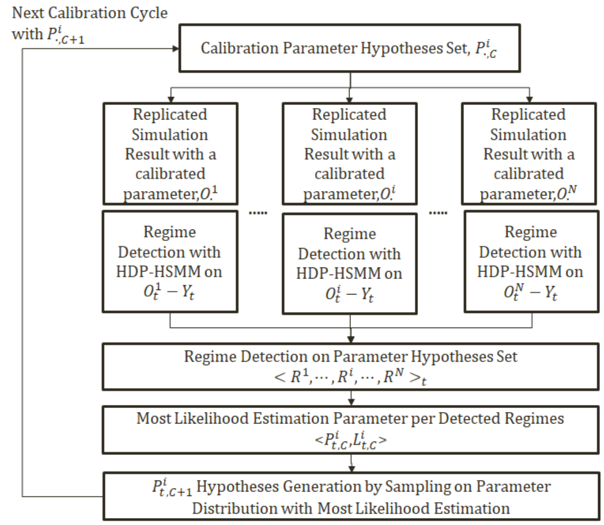

Fig. 2. Overview on the algorithm of automatic parameter calibration

A. Algorithm Overview of Automatic Parameter Calibration

This section provides the overview of the dynamic parameter calibration. Figure 2 illustrates the flowchart of the algorithm execution per each parameter updates. The introduced algorithm iteratively updates the calibrated parameters by sampling. Here, we define the parameter to be calibrated as $P^i_{t,C}$. C denotes the calibration iteration index, and i denotes the index of a single setting on the calibrated parameter, which suggest that we handle multiple parameter calibration candidate for each update iteration. Finally, t suggests the simulation timestep that the calibrated parameter value will be used. Hence, $P^i_{\cdot,C}$ is the i-th scenario of the calibrated parameter for the C-th update iteration over the entire simulation period because . means the whole period.

The algorithm starts with a set of calibrated parameters, $P^i_{\cdot,C}$, generated from the previous parameter update cycle or the arbitrarily chosen values for the first update iteration. Each setting scenario is iterated for R replicated times to find the statistically stable output values, $O^i_{\cdot}$. Then, the algorithm compares the simulation output and the validation target which is defined as $Y_{\cdot}$. Particularly, the deviation from Y to O^i is calculated for each timestep, so the deviation could be dynamically clustered as regimes. These regimes differentiates the well-fitted sequences and the far-deviating sequences when we examine $O^i_t - Y_t$. The well-fitted regimes should be calibrated with marginal updates, and the far-deviating regimes should be further updated.

The fitness is measured by the likelihood, L^i_t, of observing a target validation value, Y_t at time t given the distribution of O^i_t. Here, we assume the distribution of O^i_t as a normal distribution, and we infer the mean and the standard deviation with the replicated results with the same parameter, $P^i_{t,C}$.

Then, we calculate the likelihood, L_t^i, of observing O_t^i, and we perform the dynamic clustering with a regime detection model.

The detected regimes are represented as R^i per each parameter setting of $P^i_{\cdot,C}$ for the whole simulation period. Some of the regimes will have high average likelihood, and others will have low average likelihood. These regime detections are different by the parameter settings of $P^i_{\cdot,C}$, so we merge the regime detection results, R^i, to produce a single regime detection, $R_t =< R_t^1, \ldots, R_t^i, \ldots, R_t^N >$. This merged regime detection tells which parameter setting is more likely at a certain merged regime. We further discuss this regime detection algorithm and processing in Section 3.B.

Given a single merged regime, what we observe is a set of $< P_{t,C}^i, L_t^i >$. We infer the parameters of the underlying distribution of this observations. Particulaly, we treat $P_{t,C}^i$ as the sampled value of the distribution, and we consider L_t^i as the weight of the sampling. Afterward, we can select a distribution, i.e. Beta distribution for the ranged parameter in this paper, and we estimate the necessary parameters for the selected distribution. Finally, the fitted distribution of the parameter for the regime is utilized to generate the parameter for the next cycle. The generation could use a static selection at the key points of the distribution, i.e. the mean, the one-sigma deviations from the mean, etc, or the generation could rely on the sampling of the fitted distribution. We further discuss this parameter hypotheses generation in Section 3.C. The generated parameter hypotheses are fed back to the start of the calibration cycle as $P^i_{\cdot,C+1}$, and this finally conclude the overall description of the iterated dynamic parameter calibration.

B. Regime Detection for Calibration

The regime detection is finding a set of dynamic clusters over the simulation period, and the dynamic clusters should show a consistent pattern of deviation, $O_t^i - Y_t$. A traditional approach to this problem is the hidden Markov model (HMM). The Baum-Welch algorithm is an expectation-maximization algorithm of HMM, and the latent random variable becomes the state varaible of the HMM. The shortcoming of the Baum-Welch algorithm is the prior determination of the cluster number, K, and this is difficult because the dataset for dynamic clustering changes over the parameter settings per each cycle. Therefore, it is hard to expect a human intervention, such as setting K, in the middle of the calibration iteration. In this setting, a simplest solution is determining K as a fixed setting and accepting a parameter bias induced by the fixed parameter. This simple solution is applied to our second experimental case in Section 4.B. when the tested simulation is too complex; and when the additional variance from the calibration does not increase the validation level. However, we used a further complicated solution to our first experimental case in Section 4.A. when we use a non-parameteric version of HMM.

Hierarchical Dirichlet process-hidden Markov Movel (HDP-HMM) is a non-parameteric version of HMM, and we used HDP-HMM for the first experimental case [11]. HDP-HMM does not set the cluster number, K, but it requires the cluster creation hyper-parameters, such as γ, α, H, which is called as the table creation in the terminology of the Gibbs sampling inference of HDP-HMM. Formally, HDP-HMM is specified by the below generative process defining the probabilitstic causality between the random variables.

$$\begin{aligned}
&\beta \sim GEM(\gamma) \\
&\phi_i \overset{\text{i.i.d}}{\sim} DP(\alpha, \beta), \theta \overset{\text{i.i.d}}{\sim} H, i = 1, 2, \ldots \\
&R_t^i \sim \phi_{R_{t-1}^i} \\
&(O_t^i - Y_t) \sim f(\theta_{R_t^i})
\end{aligned} \tag{1}$$

In Equation 1, HDP-HMM is adopted to the dynamic parameter calibration setting by introducing R_t^i and $O_t^i - Y_t$ in the generative process. We can use a Gibbs sampling algorithm to infer ϕ and θ to cluster the deviation between the simulation output and the validation target. We assumed f to be a multivariate normal distribution because the simulation output could be multi-dimensional. Because of this assumption of the normal distribution, θ includes the mean vector and the covariance matrix, which are generated from the base distribution, H. A typical approach would be using the Normal-Inverse-Wishart distribution, but we fixed the mean vector to be zero and the covariance matrix to be an identity matrix with a hyper-parameter of σ. Finally, the hyper-paraemters tuned experimentally are γ, α, and σ.

C. Parameter Calibration Hypotheses Generation

After we detected the regime per each parameter setting, $P^i_{\cdot,C}$; and after we merged the regime detections into a single regime by concatenating as $R_t =< R_t^1, ..., R_t^i, ..., R_t^N >$, we generated the calibrated parameter for the next iteration. The detailed steps of the hypothesis generation is emureated in the below.

1) Identify a unique set of regime after merge : an element of this unique set is a tuple of merged regime detection of $R_t =< R_t^1, ..., R_t^i, ..., R_t^N >$ which is a distinct tuple across the simulation period, $0, ..., t, ..., T$.
2) Collect observations with the same regime in the unique set : the observations are the pair of $< P_{t,C}^i, L_t^i >$ with the same R_t across the parameter settings index by i.
3) Estimate the distribution parameters of the simulation parameter and its likelihood : the collected observations are treated as a sampling of $P_{t,C}^i$ with L_t^i weight. We assume a beta distribution, $Beta(\alpha, \beta)$, if the simulation parameter is ranged. Then, we estimated α and β of the $Beta$ distribution by utilizing the weighted samples.
4) Sample the next parameter hypothesis : Now, we have a fitted distribution for the simulation parameter, and we have two strategies in generating the hypotheses.
    a) Random sampling of $P_{t,C+1}^i$ from the simulation parameter distribution, i.e. $Beta(\alpha, \beta)$
    b) Mode sampling of $P_{t,C+1}^i$ from the simulation parameter distribution, i.e. $Beta(\alpha, \beta)$, by selecting the mean, the one-σ deviations, and so forth.

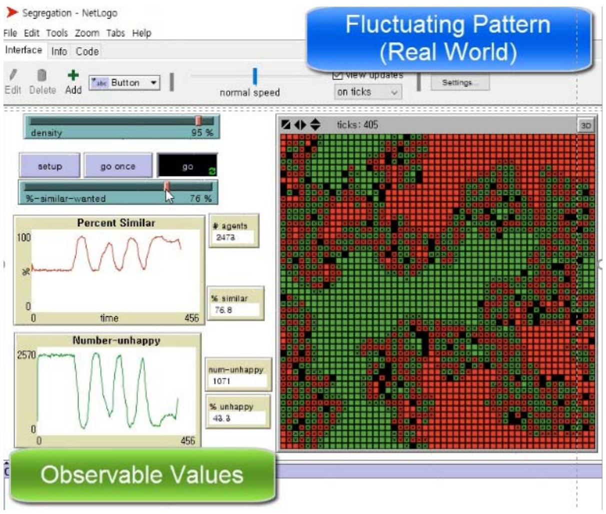

Fig. 3. Segregation model with fluctuating latent parameter, *Percent Similar*, and its simulation results, *Number-unhappy*

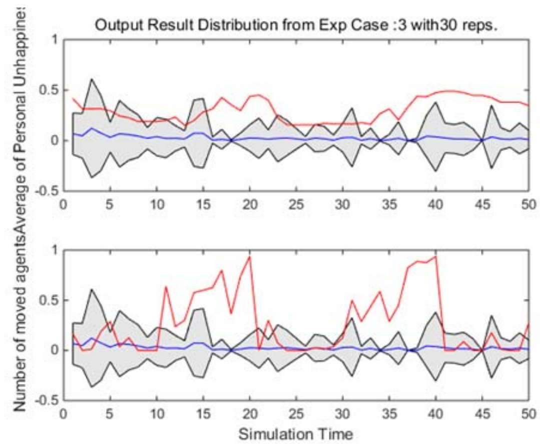

Fig. 4. Deviation from validation target data to simulation result data in two result dimensions: Number of moved agents and average of personal unhappyness

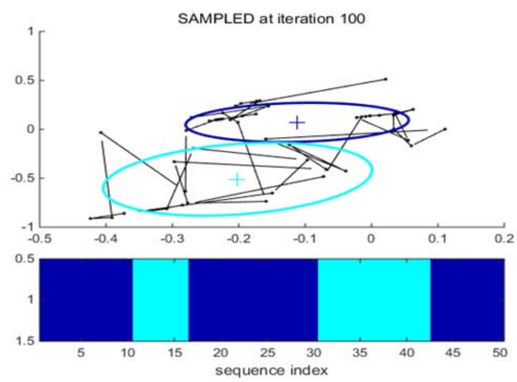

Fig. 5. Regime detection result from the deviation of validation target and simulation result with a specific parameter setting. (Upper) Two axes matching the deviations on two simulation result dimensions, (Bottom) Regime detection result over the simulation period

IV. Result

We applied our calibration algorithm to two simulation models. The first model is a hypothetical model with a synthesized scenario, which is used to verify the feasibility of the alogrithm. The second model is a simulation model to reflect the real estate transactions in the real world. This is an actual calibration study to increase the validation level.

A. Test Case 1: Schelling's Segregation

1) Model Description: Thomas Schelling's segregation model, see Figure 3, is one of the most well-known agent based models, and we choose this model to synthesize the calibration scenario because a realistic model with a real world validation dataset cannot drill down the validity of the calibrated parameters. We calibrate a dynamic parameter, *Percent Similar*, over the simulation period when we have a validation target of *the average of personal unhappyness* and *the number of moved agents* with a predetermined *Percent Similar* that is undisclosed to the calibration model.

2) Calibration and Validation: This calibration starts from simulating three parameter settings that are initially and arbitrailly generated. We replicated the simulation for 30 times to calculate the mean and the standard deviation of the results, which are illustrated in Figure 4. This figured showed a period of well-fitted period, which shows close distance between the mean blue line and the validation target red line.

After calculating the deviation of two simulation results and validation dataset, we performed HDP-HMM to detect the regimes as Figure 5. Since the two dimensions need to be close to zero mean of the deviation, Figure 5 suggests a blue cluster for a relatively well-fitted regime and a cyan regime for better calibration.

We have three parameter settings, and Figure 6 shows three merged regimes and the likelihoods by each parameter setting and timestep. Now, we found a set of unique merged regimes; and the regime-associated parameter and likelihood pairs. Afterwards, we estimate the *Beta* distribution parameter of the simulation parameter, *Percent Similar*. Figure 7 shows the mean value for each time step by comparing the mean value as the grey line to the undisclosed parameter as the red line. The visualized calibrated dynamic parameter is captured at 19th iteration out of 20 iterations.

Finally, we show the validation improvement over 20 calibration iterations in Figure 8. The absolute values on two deviations corresponding to two simulation outputs become continuously lower, and this demonstrate the validation target is regenerated closely by the calibrated simulation parameter.

B. Test Case 2: Agent Based Model for Real Estate Market

1) Model Description: The purpose of this agent-based model is to analyze the impact of changes in housing policy on the housing market and the society. From a macroscopic

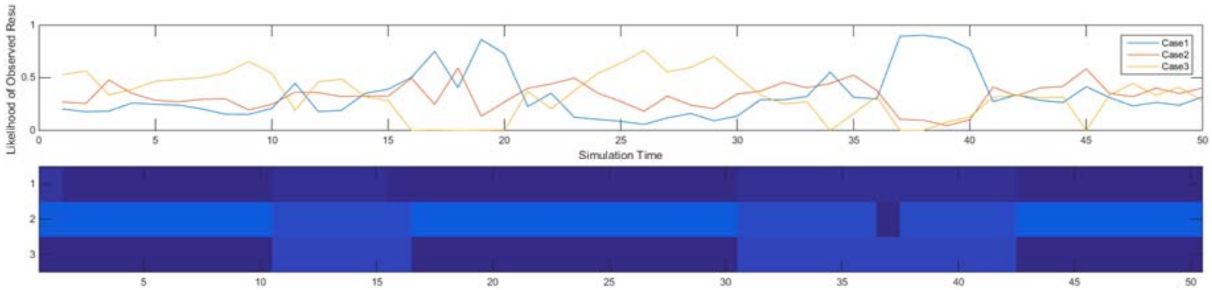

Fig. 6. (Upper) Likelihood of observing validation target given a set of simulation replications with a specific parameter settings, (Bottom) Merged regime detection result across parameter settings (currently, total three settings were used)

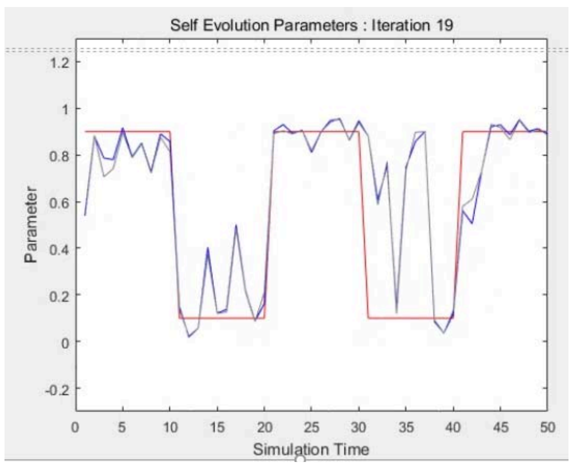

Fig. 7. Parameter calibration result, (Red Line) true target parameter value, (Grey Line) parameter value after calibration

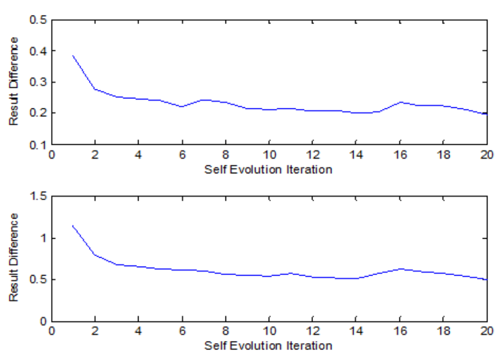

Fig. 8. Absolute result difference between validation target and calibrated simulation result on two simulation output dimensions (Upper) Average of personal unhappyness (Bottom) Number of moved agents

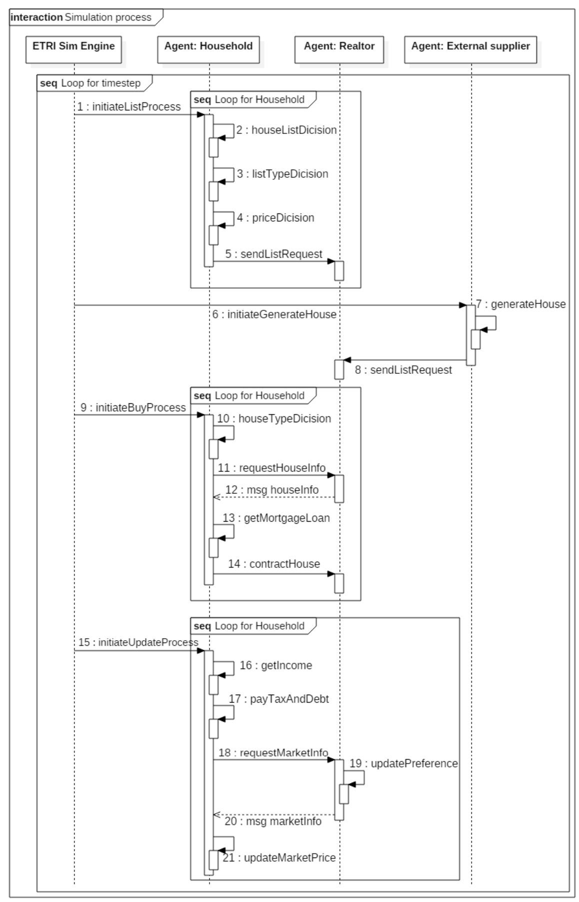

Fig. 9. Sequence diagram to describe the interactions between agents of real estate markets

point of view, we can analyze price and liquidity dynamics in the housing market. Also from a micro perspective, we can see how the financial situation of households changes according to housing policy. This model is currently in line with the Korean housing market. The properties of the agents are based on the Korean census data, and many economic variables using in the simulation model come from the real data of Korea. The model consists of three kinds of agents; household, external supplier and realtor. There are N households in the simulation model. Households use their own savings and loan services to meet their residential needs. Households make decisions about buying, selling and leasing houses. The external supplier

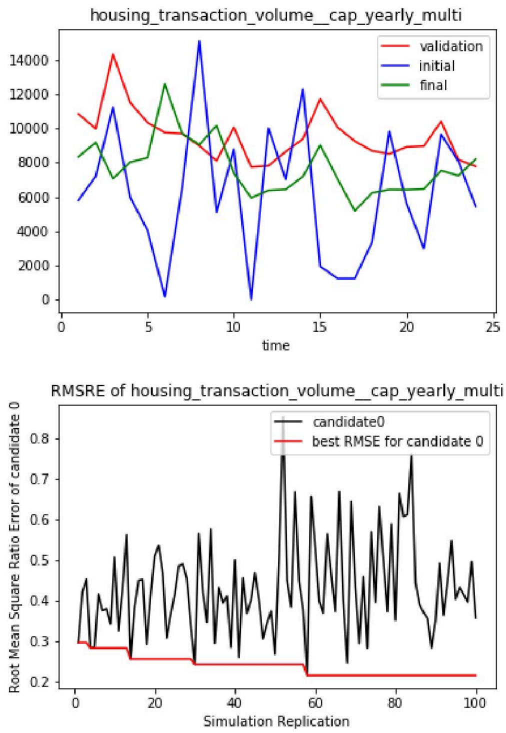

Fig. 10. (Upper) One simulation output dimension from teal estate agent based model. The final calibration in green becomes closer to the validation target in red compared to the initial setting in blue. (Bottom) The optimized RMSE in the simulation output as the parameter search iterations progress

serves to supply new houses and there is only one in the simulation model. Lastly, the realtor registers the house on the housing market that owner want to sell or lend, and provides housing information to the agent who wants to buy or lease the house. Figure 9 visualizes the interactions between agents. The simulation model is executed by iteratively executing the list process, the buy process, and the update process. First, the list process is the process of listing the houses in the market that the owner want to sell or lend. Second, in the buy process, the households make decisions about purchasing or leasing a house. They first determine the preferred type of housing transaction and house type. Then they request information of houses suitable for their preference to the realtor.Lastly, in the update process, update savings of the households and market price of the houses. In the case of savings, the households have their own earnings based on census data. Then, they save the remaining portion of income, excluding consumption, taxes, rent fee, and loan repayment. On the other hand, the house price is updated by using the preference of similar houses in the previous buy process and the market condition, for example inflation rate.

2) Calibration and Validation: Given the limited paper space, we cannot enumerate the details of the calibration result as we illustrated in the segregation model. Additionally, unlike the synthesized case, we do not know the ground truth of the parameter to calibrate in a real-world case. Therefore, we can only evaluate our model by comparing the validation target and the simulation result as illustrated in Figure 10. Our calibration study in the real world case shows that the search becomes very noise as shown in the bottom of Figure 10. However, the found well-fitted parameter setting does produce more validated simulation result with less RMSE.

V. Conclusion

This paper proposes an automatic calibration mechanism for simulation models with data-driven methdologies. Particularly, we suggest how to adopt the non-parameteric regime detection algorithm of HDP-HMM; and how to sample a further calibrated parameter setting per detected regimes. Finally, we put these models and heuristics into a iterated algorithm of parameter calibration. Our study were tested in two different scenarios: Schelling's segregation and a case study of agent based simulation on real estate business, and both studies show the improved validation levels.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2016-0-00094-003, Development of Predictive Analysis Technology on Socio-Economics using Self-Evolving Agent-Based Simulation embedded with Incremental Machine Learning).

References

[1] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[2] O. Balci, "Principles and techniques of simulation validation, verification, and testing," in *Proceedings of the 27th conference on Winter simulation*. IEEE Computer Society, 1995, pp. 147–154.

[3] P. Davidsson, "Agent based social simulation: A computer science view," *Journal of artificial societies and social simulation*, vol. 5, no. 1, 2002.

[4] N. Gilbert, "Emergence in social simulation," in *Artificial societies*. Routledge, 2006, pp. 134–143.

[5] K. M. Carley, "Computational approaches to sociological theorizing," in *Handbook of sociological theory*. Springer, 2001, pp. 69–83.

[6] P. Windrum, G. Fagiolo, and A. Moneta, "Empirical validation of agent-based models: Alternatives and prospects," *Journal of Artificial Societies and Social Simulation*, vol. 10, no. 2, p. 8, 2007.

[7] B. Park and H. Qi, "Development and evaluation of a procedure for the calibration of simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1934, pp. 208–217, 2005.

[8] T. Toledo, M. Ben-Akiva, D. Darda, M. Jha, and H. Koutsopoulos, "Calibration of microscopic traffic simulation models with aggregate data," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1876, pp. 10–19, 2004.

[9] J. Hourdakis, P. Michalopoulos, and J. Kottommannil, "Practical procedure for calibrating microscopic traffic simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1852, pp. 130–139, 2003.

[10] V. Nannen and A. E. Eiben, "A method for parameter calibration and relevance estimation in evolutionary algorithms," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 183–190.

[11] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An hdp-hmm for systems with state persistence," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 312–319.