

Analyzing Agent-based Models using Category Theory

Rahmatollah Beheshti and Gita Sukthankar

Department of EECS

University of Central Florida

Orlando, Florida 32816, USA

Email: {beheshti@knights, gitars@cs}.ucf.edu

Abstract—Agent-based models are a useful technique for rapidly prototyping complex social systems; they are widely used in a number of disciplines and can yield theoretical insights that are different from those produced by a variable based analysis. However, it remains difficult to compare the results of two models and to validate the performance of an agent-based simulation. In this paper, we present a case study on how to analyze the relationship between agent-based models using category theory. Category theory is a powerful mathematical methodology that was originally introduced to organize mathematical ideas according to their shared structure. It has been successfully employed in abstract mathematical domains, but has also enjoyed some success as a formalism for software engineering. Here we present a procedure for analyzing agent-based models using category theory and a case study in its usage at analyzing two different types of simulations.

Keywords—category theory; agent-based modeling; Markov-chain Monte Carlo

I. INTRODUCTION

Agent-based modeling has been used successfully for studying many types of social and biological phenomena. Although the gold-standard test for an ABM is comparing its predictions to real-world data, often paucity of data can eliminate this form of comparison. More commonly, domain experts can be used to guide the modeler during the creation of the model and tuning of parameters. However, comparing one model to another remains a difficult challenge, particularly because it is often problematic to formally specify many types of agent-based models.

In this paper, we present a case study on the use of category theory to formally specify and analyze agent-based models. The power of category theory is that it can be used to express different types of systems in a common language. It was originally introduced in order to handle problems in algebraic topology and homology theory [1]. Category theory enables one to abstract a formal system by eliminating superfluous details. By mapping a problem to a seemingly unrelated problem in another domain, it becomes possible to leverage known proofs and solutions from the original domain.

The power of category theory mainly comes from its focus on relations among the objects rather than the objects themselves. Historically, most of structures defined in category theory were defined in order to study and represent

complex structures in a consistent way. Healy et al. [2] use the following analogy to illustrate the role that category theory could play in studying different disciplines. Imagine a scientist viewing an electrical circuit and a chemical compound. At first glance, they might appear to be very different structures, but a deeper look reveals that chemical bonds are also electrical in nature. A common meeting ground between electricity and chemistry can be found within the abstractions of physics: quantum states and the large-scale static/dynamic properties of electrons. These abstractions allow the scientist to define the relationship between electrical circuits and chemical compounds, transfer insights from one discipline to another, and study electrochemical reactions. Agent-based models are often used to encode discipline-specific ideas from psychology, sociology, or biology on the function of a complex system [3]. We believe that representing these models in category theory is the key to understanding the relationship between multiple agent-based models of the same system. Category theory empowers us to create mappings between the models and understand their operation in a functional way, rather than simply comparing the predictions of the simulations.

This paper presents a case study in the usage of category theory for comparing different variants of an urban simulation system, designed to study traffic congestion and parking scarcity on a large university campus. In this work, category theory is initially used to represent our agent-based model and a Markov-chain Monte Carlo sampler that can be combined with survey data to estimate quantities of interest. We describe how category theory can be used to represent the relationship between the two models and how insights from the category theory representation can facilitate the creation of hybrid modeling methods.

II. RELATED WORK

Agent-based models are a popular modeling and simulation technique due to their ease of construction [3]. The modeler simply defines a population of agents with specific properties, plus a set of rules governing the agents' behavior and decisions. It is relatively simple to rapidly prototype a complex system with emergent behaviors, even without a formal specification or complete knowledge of the system dynamics. ABMs have been applied to a range

of interesting real-world problems ranging from modeling people's transportation selections to simulating the response of an organ's cells to a bacterial attack [4]. Yet, the lack of mathematical formalism can make the results of ABMs hard to validate and also render them difficult to reproduce. The results of a simulation study can vary considerably by changing the range or even the step size of just one or two variables [5]. According to [6], most of works based on ABM, with the exception of a few classic models, have never been replicated by anyone but the original developer. For the case studies described in this paper, we use an urban transportation model that couples an activity-oriented microsimulation with path planning. Each agent represents a student on the UCF campus, and the population was created to match the data from a survey on student transportation, parking, and dining preferences.

As our second modeling methodology, we select a method drawn from the statistics community, Markov-chain Monte Carlo estimation. Markov-chain Monte-Carlo (MCMC) describes a family of methods for performing Bayesian inferences using stochastic simulation. It has been widely used in a variety of scientific [7] and engineering modeling applications [8]. MCMC allows us to draw samples from a distribution $\pi(x)$ without having to know its normalization. Having these samples, it is possible to compute any quantity of interest about the distribution of x , including confidence regions, means, standard deviations, and covariance [9].

In this paper, we attempt to relate these two families of modeling methods using the abstraction language of category theory (CT). Category theory has been successfully used in several branches of mathematics, including geometry, algebra, and logic [10], [11]. But CT can also be used by researchers to describe physical and social systems. A historical review of CT applied to physics abstractions can be found in [12]. Coecke [13] asserts that category theory should become part of the daily practice of the physicist. Recently, Sallach [14] illustrated the benefits of categorical analysis within the social sciences by using CT to explicate several well known social theories. For instance, he shows how the equivalence and duality relations (structures in CT) can be used to explain Pareto's theory of the circulation of elites.

There has been some use of category theory within software engineering in which CT is used as alternate formal specification language. For example, in [15] the author describes how the concepts of category theory can guide the design of a programming language to avoid anomalies in the interaction of implicit conversions and generic operators. The rigorous mathematical formalism of CT can empower software developers to reason about structures within their code. In addition, it provides an exact notion of modularity and composition. Another major application area of category theory within computer science is data analysis. As an example, [16] formally defines information fusion in category the-

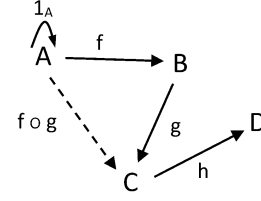


Figure 1: A simple category containing objects A, B, C, D , morphisms f, g and h , identity arrow 1_A and composite arrow $f \circ g$

ory, and then shows how one can carry out formal reasoning about information fusion systems. Within machine learning, specific categorical constructs were applied to determine neural structures for the re-design of a neural network [2]. In this paper, our aim is twofold: 1) to use category theory to provide a formal representation for our ABMs and 2) to use the mapping between multiple models to motivate the development of new hybrid modeling techniques. In the next section, we provide the reader with some background in category theory.

III. A CONCISE INTRODUCTION TO CATEGORY THEORY

In order to reach the point that we can define our desired representation using category theory, we need to briefly introduce the required structures. For a detailed overview of category theory elements the reader is referred to [1] and [17]. The basic structures that are defined in category theory are the category itself, arrow, and functor.

- A category \mathbf{C} consists of: 1) a set of objects (A, B, C, \dots), 2) a set of arrows (f, g, h, \dots) also known as morphisms, 3) a way to compose arrows (composed arrows are also associative), 4) identity arrows. Each arrow has a unique source or domain and a unique target or codomain. Figure 1 shows a simple category containing objects A, B, C, D and the arrows f, g and h . The identity arrow for object A and composite arrow of f and g are shown in this figure.

- A functor $\mathbf{C} \rightarrow \mathbf{D}$, shown by F , is a mapping from objects to objects and arrows to arrows of category \mathbf{D} . For the objects and arrows, we define $F(f : A \rightarrow B) = F(f) : F(A) \rightarrow F(B)$. In addition to domains and codomains, functors will preserve identity arrows and composition. Figure 2 shows a functor mapping category \mathbf{C} to category \mathbf{D} .

As was mentioned earlier, the focus in category theory is on relations rather than objects. Accordingly, various structures relating different types of relations at multiple levels are defined. Arrows show the relations among objects of a category, and functors show relations among categories. The relation among the functors is also shown by natural transformations. One could imagine natural transformation doing the same to two functors between two categories,

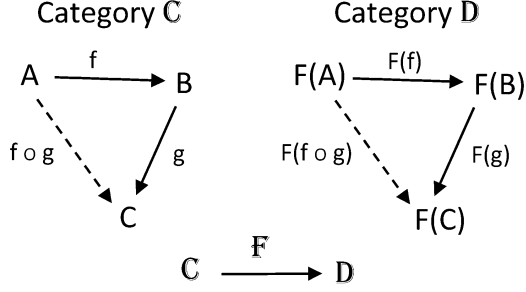
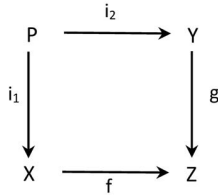


Figure 2: A functor going from category \mathbf{C} to \mathbf{D}

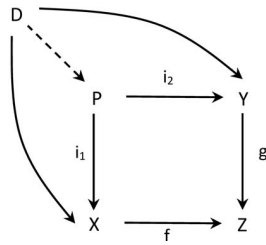
as what functors do to the objects and morphisms of two categories.

- Another key concept in category theory is universal property. It is hard to come up with a concise and exact definition of universal property, and it is usually defined through some other structures, so the reader is referred to the introduced references for a formal description [17]. Informally universal property refers to those set of properties that apply to all of objects in a category, and is the best and most effective set of properties they share; the idea of universal property directly relates to optimization in a system. Many ideas in category theory are based on the universal property concept such as limits, initials, products and their dual. The dual of each structure in category theory is constructed by reversing all of the existing morphisms.

- The other definition we need is the pullback structure. In the square



the morphisms i_1 and i_2 plus the object P are called the pullback of morphisms f and g . If the pullback is a universal property, there should be a unique morphism between object P and any other object like Q that is the domain of two morphisms to X and Y .



- The last structure that will be introduced here is adjoint functors. Since the main contribution of this work is

Relation	Equality	Isomorphism	Equivalence	Adjunction
Meaning	$\mathbf{C} = \mathbf{D}$	$1_{\mathbf{C}} = GF$ and $FG = 1_{\mathbf{D}}$	$1_{\mathbf{C}} \cong GF$ and $FG \cong 1_{\mathbf{D}}$	$1_{\mathbf{C}} \Rightarrow GF$ and $FG \Rightarrow 1_{\mathbf{D}}$

Table I: Hierarchy of relations between two categories \mathbf{C} and \mathbf{D} , in terms of equality. F is a functor going from \mathbf{C} to \mathbf{D} , and G is dual of F . \cong shows isomorphism and \Rightarrow shows natural transformation relations.

presented using this structure, we will provide more details about it. Category theory excels at expressing weaker types of equality in a mathematical language. Imagine we have two categories \mathbf{C} and \mathbf{D} , and two functors F and G between them, as Figure 3 shows:

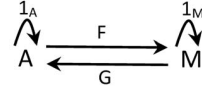


Figure 3: Two categories \mathbf{C} and \mathbf{D} , and functors F and G between them

A hierarchy of relations could be defined between these two categories as shown in Table I. As the table illustrates, equality refers to the classic relation of two items that is quite rigid. It simply means they are the same ones. Isomorphism is more lenient than equality and states that going from category \mathbf{C} to category \mathbf{D} and then returning (GF) is equivalent to remaining at \mathbf{C} ($1_{\mathbf{C}}$). The same thing exists for category \mathbf{D} . Descending the hierarchy, we reach an equivalence which says that going from \mathbf{C} to \mathbf{D} and returning (GF) is isomorphic to the identity functor or $1_{\mathbf{C}}$. Finally, adjunction is even weaker than the other relations. It says that there exists a natural transformation from the path starting from \mathbf{C} to \mathbf{D} and returning to \mathbf{C} (GF), to the identity functor ($1_{\mathbf{C}}$). A similar natural transformation exists for the other case.

IV. APPLYING CATEGORY THEORY

To apply category theory, the first thing we need to do is to define the required categories. We define category \mathbf{A} as representing an arbitrary agent-based model, and category \mathbf{M} as representing an arbitrary Markov-chain Monte-Carlo model. The challenging aspect of using category theory is often showing that the desired structures can be considered a category. In order to show this, we need to show that these structures have all of the properties listed in the definition of a category. We will return to this point later in this section after introducing the elements of categories \mathbf{A} and \mathbf{M} .

The approach we are going to use here to describe these categories is partly based on the representation approach described in [18]. This method was used to show the agent-based modeling in CT language and is mainly based on ideas

from inverse theory, which is the process of finding the best values for the parameters associated with an assumed model based on the observed data [19]. Inverse theory is itself an extensive and thorough theory. Here, we just need a couple of elementary ideas from inverse theory to define the objects in our categories. The purpose of employing ideas from inverse theory is to define formally what is meant by model and data in our representation. The forward problem in inverse theory relates to the problem of predicting data based on the description of the model parameters. Using elements from category theory language, the forward function can be represented by a morphism from object M to object D as follows:

$$F : M \longrightarrow D \quad (1)$$

Similarly, the inverse problem can move from data to model, as shown here:

$$F' : D \longrightarrow M \quad (2)$$

Additionally, another object namely the universal object, U , can be defined. This object refers to all of the existing information about the system. Some portion of this information is assumed to be known through available data, and the rest will be (partially or totally) produced through the modeling technique (e.g., ABM or MCMC).

The process of moving from model to data or from data to model can be also studied on a Bayesian basis. Hence, two new objects related to the conditional probability of objects M and D can be added to the objects defined so far: $M|D$ and $D|M$. $M|D$ (model given data) refers to the addition of data to the model, or the situation of inferring the model from data, and $D|M$ (data given model) represents the opposite process.

The next step is to define the categorical representation of an ABM and MCMC model. These categories are presented using the pullback structure introduced earlier, and are shown in Figure 4. In this case, there is a commuting square that according to the definition of pullback should be universal, meaning that it should be the *best* among all similar squares. In CT language, this means *given any other one there should be a unique morphism/factorization to M* . M is the vertex with two projections in the square. Here, *any other one* refers to any other square which also has the two morphisms $M|D \longrightarrow U$ and $D|M \longrightarrow U$. More exactly, this can also be imagined by considering different observations or different sets of data. This is shown by D_1 to D_n in Figure 5. The reason why a pullback was selected to present this structure is the inherent universal property in pullback. If we assume that only one correct model exists—which in theory is a valid assumption regarding a system—then the model exactly plays the role of an universal object in a categorical structure. That is to say no matter which modeling technique we use, as long as the system is the

same, there exists a unique model showing the system. In terms of category theory, universal object is the *best* or *most efficient* object, considered as a factor of other objects¹. Four types of morphisms are presented in this structure. Type f which shows the mapping from the conditional knowledge, $D|M$ and $M|D$, to the universal knowledge, U . This is shown in Figure 4 by f_1 and f_2 . Type i refers to the model transition to the conditional knowledge. Type j shows the morphisms from data (observation) to the conditional knowledge. Finally, type p shows the unique² morphism that should exist between data D and model M . This denotes the probabilistic relation that exists between data and model in any Bayesian domain. In other words, if we look at the ABM and MCMC as both sample generator techniques, the probability that is obtained by the population of samples represents the morphism p defined between data object D and M .

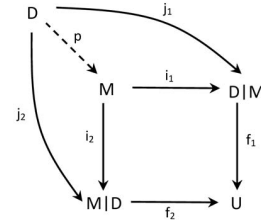


Figure 4: The proposed categorical representation for an arbitrary ABM or MCMC model

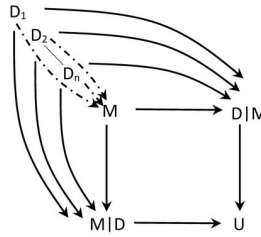


Figure 5: There is a unique mapping between the object model, M , and each object corresponding to the different observed datasets, $D_1 \dots D_n$

Armed with these definitions, we can verify the compliance of the proposed categorical structures with the formal definition of category in CT. The suggested structure has objects and morphisms defined; the morphisms are associative. For simplicity, identity arrows for the objects are not shown in the figures. Two types of composed relations could be imagined in this structure as shown on either side of the

¹Not to be mistaken with the object U we defined for the categories. Object U , as its definition shows, just represents the universal knowledge about the problem. It is an ordinary object in the category we defined. It does not possess any universal property which is a separate concept.

²According to the definition of pullback, there should be a unique morphism from each object D to object M on the corner of square.

set of equalities in Equation 3. In order to have associativity property, the following equalities must hold. Since we have only one model object, these equalities exist among the composite relations. Here, \circ operator shows the composition of two morphisms.

$$\begin{aligned} f_1 \circ (i_1 \circ p) &= (f_1 \circ i_1) \circ p \\ f_2 \circ (i_2 \circ p) &= (f_2 \circ i_2) \circ p \end{aligned} \quad (3)$$

We defined our categorical structure in such way that it could be used to describe both methods at the same time, so no other category is required. This, by itself, shows the similarities between these two different methods. Now, we can describe the formal relationship between the two categories. The way that the two categories are defined allows us to observe that an adjunction exists between the two categories A and M , which can be represented by the same structure shown in Figure 4. This way, it is simple to see that there exists a left adjoint functor from category A to category M , and a right adjoint functor from category M to category A . This is shown in Figure 6.

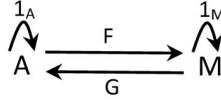


Figure 6: The adjunction between A , the category showing an arbitrary ABM, and M , the category showing an arbitrary MCMC model

In order to show this, we need to prove that there exists a pair of functors and a pair of natural transformations between the two categories. The first part is trivial. Since for each object/morphism in category A , there exists a corresponding object/morphism in category M , a functor from category A to category M exists. The same justification can be used to show that a functor from category M to category A exists. For the second part, we can show that two natural transformations, η and ϵ exist. These two are shown in Equation 4. Showing the existence of these two is again trivial. Since the functors essentially map the same type of objects and morphisms between two categories A and M , η and ϵ exist between functors F and G .

$$\begin{aligned} \eta : 1_A &\Rightarrow GF \\ \epsilon : FG &\Rightarrow 1_M \end{aligned} \quad (4)$$

One question that might come to mind is that among the four types of relations introduced in Table I, why adjunction is chosen to show the relation between categories A and M ? To answer this, we should note that in all of the other three relations some sort of ‘being the same’ exists by definition, but in adjunction, we generally do not care about being the

same. Instead, we focus on the interesting relations between the two categories. Additionally, choosing adjunction for our purpose does not prevent the usage of other relations, and does not state that the others cannot exist at all. What is important is that it enables us to reach to our goal, which was to formally represent the relation among the ABM and MCMC models.

V. INSIGHTS FROM CATEGORY THEORY

Thus far, we have proven that the two methods can be shown to be equal (up to natural transformation) in terms of category theory, i.e. the weakest equality. Hence it is possible to combine the two methods to produce a hybrid modeling methodology that builds on the strengths of both models. Theoretically any two or more methods that function in a shared domain and can be used in sequential manner could be considered as candidates for building a hybrid method. But, the difficult part is to show why this hybrid method is valuable. Using category language, we can evaluate the feasibility of different models for hybridization.

Our hybrid model (ABM-MCMC) uses the population of samples generated by the agent-based model to initialize the proposal distribution for the MCMC estimator [20]. In the categorical representation of ABM, prior knowledge (data) shown by D determines how the agent-model should be constructed. Then, through an iterative process the data determines the model by producing samples while agents remain active in the system. The Bayesian approach of representing this process helps us to understand the similarity between ABM and MCMC better. The samples produced by the agent-based model become the data within the MCMC component. This hybrid method is able to resolve the proposal distribution problem of MCMC methods, while possessing greater verification possibilities than the ABM alone.

VI. RESULTS

In order to validate the performance of our hybrid method, two case studies are presented in this section. The first case study is related to an urban modeling problem while the second describes a marketing scenario. In both application domains, we show that the hybrid method (ABM-MCMC) outperforms either ABM or MCMC alone. Both agent-based models presented here are implemented in Netlogo [21], and the MCMC component of the hybrid method is run using the MCMCpack package in R [22]. Implementation details about these case studies are omitted from this section since they are irrelevant to the main thrust of the paper and can be found in the original descriptions of the ABMs.

A. Urban transportation simulation

This case study extends our earlier work described in depth in [23] and [20]. The code for the urban ABM can be

freely accessed online³. The aim of this project was to model the transportation patterns of students at the University of Central Florida (UCF). The data for this study was collected through an online survey. A detailed agent-based model was created based on the survey data of students' housing, dining, and transportation preferences. The simulation can be used to perform analyses of traffic patterns, building occupancy and parking usage. Here, we specifically present the results related to student parking usage for comparing the different models since it was possible to obtain ground truth data for this quantity from UCF Parking Services.

For ABM-MCMC part, the data samples generated by the agent-based model showing the location of each agent were used to initialize the proposal distribution. Additionally, in order to test the MCMC method alone, the MCMC toolbox for MATLAB [24] was used in combination with the original survey data. In this toolbox, the prior distribution is simply assumed to be in form of Gaussian distribution. Figure 7 shows the results obtained using each of these three modeling approaches.

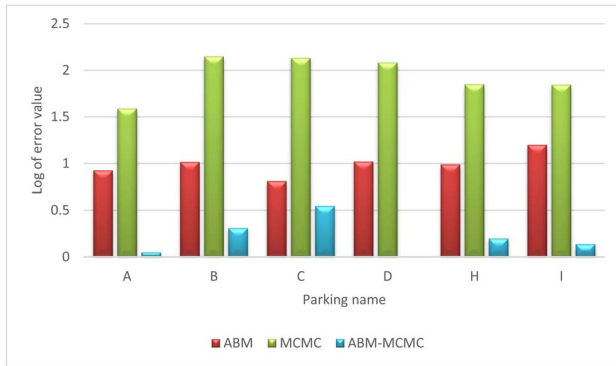


Figure 7: The log of the difference between the number of cars predicted by each method and the numbers from empirical data. Shorter bars show a smaller deviation between the model prediction and the actual data. Our hybrid method, ABM-MCMC, outperforms the parent methods in all cases.

As the figure shows, the hybrid method outperforms the original methods in terms of accuracy of prediction.

B. Marketing analysis

The second case study is based on an agent based model published by Rand et al. [25] in which the agents are used to model consumer behavior. The main purpose of this work was to simulate the famous Bass model [26] published in 1969. The authors also study the role of different network structures on the market's behavior. The code and detailed documentation of the agent based model are freely available online. The Bass model describes how a population of consumers adapt to new products. This is

Product	Period of forecast	ABM-MCMC R^2	ABM R^2	Bass R^2
AC	1950-61	0.86	0.72	0.90
Bed	1950-61	0.91	0.93	0.93
Coffee	1951-61	0.77	0.74	0.69
Dryer	1950-61	0.86	0.85	0.85
Freezer	1947-61	0.64	0.60	0.47
Lawnmower	1949-61	0.93	0.93	0.89
Refrigerator	1926-40	0.63	0.61	0.76
TV	1949-61	0.13	0.19	0.07

Table II: Accuracy of ABM-MCMC in comparison with ABM and original Bass model

done by defining two type of consumers: innovators and imitators. The behavior of the model is determined by three parameters: degree of innovation (p), degree of imitation (q) and market size (m). The same parameters are used for both the Bass model and the agent-based one. The R^2 correlation between the empirical sale data (showing the number of units sold each year) and results from the original Bass model, Rand's paper, and our hybrid method are presented in Table II. The correlation value is computed using the RSQ function in Microsoft Excel. In order to have a larger set of samples to feed the MCMC method, we ran the agent-based model 50 times.

The hybrid method shows a slight improvement in prediction accuracy. The difference between correlation values is not significant, due partially to the fact that the amount of available empirical data in the Bass original paper is fairly small.

VII. CONCLUSION

In this paper, we illustrate how category theory can be used to formally represent two popular modeling techniques, agent-based models and Markov Chain Monte Carlo simulation. Abstractions from category theory can be used to relate the different models using adjunction and form the basis for our proposed hybrid implementation of the parent models (ABM-MCMC). To demonstrate the benefits of our hybrid model, we present two case studies, urban transportation and consumer modeling, where ABM-MCMC outperforms the original modeling methodologies.

ACKNOWLEDGMENTS

This research was supported in part by DARPA award D13AP00002 and NSF IIS-08451. The authors would like to thank Dr. William Rand for offering code, data, and help. Additionally, we appreciate Dr. David Sallach's kind attention with providing us with his in press papers and presentations.

³<https://code.google.com/p/ucf-abm/>

REFERENCES

- [1] M. Kao, N. Fillion, and J. Bell, "Jean-Pierre Marquis. From a Geometrical Point of View: A Study of the History and Philosophy of Category Theory," *Philosophia Mathematica*, vol. 18, no. 2, pp. 227–234, 2010.
- [2] M. J. Healy, R. D. Olinger, R. J. Young, S. E. Taylor, T. Caudell, and K. W. Larson, "Applying category theory to improve the performance of a neural architecture," *Neurocomputing*, vol. 72, no. 13, pp. 3158–3173, 2009.
- [3] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation," *Journal of Simulation*, vol. 4, no. 3, pp. 151–162, 2010.
- [4] B. Heath, R. Hill, and F. Ciarallo, "A survey of agent-based modeling practices (January 1998 to July 2008)," *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 4, p. 9, 2009.
- [5] M. A. Niazi, A. Hussain, and M. Kolberg, "Verification & validation of agent based simulations using the VOMAS (virtual overlay multi-agent system) approach," in *MAS&S at Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW)*. CEUR-WS, 2009.
- [6] U. Wilensky and W. Rand, "Making models match: Replicating an agent-based model," *Journal of Artificial Societies and Social Simulation*, vol. 10, no. 4, p. 2, 2007.
- [7] R. Liu, J. Tao, N. Shi, and X. He, "Bayesian analysis of the patterns of biological susceptibility via reversible jump MCMC sampling," *Computational Statistics & Data Analysis*, vol. 55, no. 3, pp. 1498–1508, 2011.
- [8] Y. Liu, Q. Wang, J. Liu, and T. Wark, "MCMC-based indoor localization with a smart phone and sparse WiFi access points," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March 2012, pp. 247–252.
- [9] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [10] F. Borceux, *Handbook of Categorical Algebra: Volume 3, Sheaf Theory*. Cambridge University Press, 1994, vol. 1–3.
- [11] S. Mac Lane and I. Moerdijk, *Sheaves in geometry and logic: A first introduction to topos theory*. Springer Verlag, 1992.
- [12] J. C. Baez and A. Lauda, "A prehistory of n-categorical physics," *Deep Beauty*, p. 13, 2009.
- [13] B. Coecke, "Introducing categories to the practicing physicist," in *What is category theory*, 2006, pp. 45–74.
- [14] D. L. Sallach, Argonne National Laboratory and University of Chicago, in press work, <http://1.usa.gov/13C2FIA>.
- [15] J. C. Reynolds, "Using category theory to design programming languages," in *Programming Languages and Systems*. Springer, 2009, pp. 62–63.
- [16] M. M. Kokar, J. A. Tomasik, and J. Weyman, "Data vs. decision fusion in the category theory framework," in *Proceedings of the International Conference on Information Fusion (FUSION)*, 2001.
- [17] S. Awodey, *Category theory*. Oxford University Press, USA, 2006.
- [18] K. A. Lloyd, "Category theory for agent-based modeling and simulation," 2010, obtained from: <http://bit.ly/12eHkhw>.
- [19] A. Kirsch, *An introduction to the mathematical theory of inverse problems*. Springer Science+Business Media, 2011, vol. 120.
- [20] R. Beheshti and G. Sukthankar, "Improving Markov Chain Monte Carlo estimation with agent-based models," in *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 2013, vol. 7812, pp. 495–502.
- [21] U. Wilensky, 1999, netLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University. Retrieved from: <http://ccl.northwestern.edu/netlogo/>.
- [22] R. D. C. Team, "R: A language and environment for statistical computing," 2012, Vienna, Austria: R Foundation for Statistical Computing.
- [23] R. Beheshti and G. Sukthankar, "Extracting agent-based models of human transportation patterns," in *Proceedings of the ASE/IEEE International Conference on Social Informatics*, Washington, D.C., December 2012, pp. 157–164.
- [24] M. Laine, 2013, MCMC toolbox for Matlab, Finnish meteorological institute. Retrieved from: <http://bit.ly/187ej96>.
- [25] W. Rand and R. T. Rust, "Agent-based modeling in marketing: Guidelines for rigor," *International Journal of Research in Marketing*, vol. 28, no. 3, pp. 181–193, 2011.
- [26] F. M. Bass, "Comments on a new product growth for model consumer durables the Bass model," *Management Science*, vol. 50, no. 12, pp. 1833–1840, 2004.