SimAnalyzer: automated description of groups dynamics in agent-based simulations

Philippe Caillou, Javier Gil-Quijano

SimAnalyzer : Automated description of groups dynamics in agent-based simulations

(Extended Abstract)

Philippe Caillou
Laboratoire de Recherche en Informatique
Universite Paris Sud – INRIA
Orsay 91405, France
philippe.caillou@lri.fr

Javier Gil-Quijano
CEA, LIST, Laboratoire Information Modeles et Apprentissage
F-91191 Gif-sur-Yvette, France
javier.gil-quijano@cea.fr

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Experimentation, Meausurement, Verification

Keywords

Complex systems simulation, multiagent systems, automated observation, automated description, clustering, value-test

1. INTRODUCTION

Multi-agents based simulations (MABS) have been successfully used to model complex systems in multiple areas. However, a pitfall of MABS is that their complexity increases with the number of agents and behaviors considered in the model. For average and large systems, different phenomena can simultaneously occur at different intermediate levels and influence each other [2]. For instance, groups of agents (flocks of birds, social groups, etc.) following similar state's trajectories may appear, evolve and disappear. To describe and evaluate the evolution of groups, the observation of global and individual variables (like in [1]) is not sufficient anymore. Moreover, because of the emergent properties of complex systems, those groups may be unexpected, or their presence may even be unnoticed because no suited variable or any other adapted observation mechanism is provided in the simulator. The significance and even the existence of groups can then be hidden by the usually huge amount of available data. In this paper we introduce the use of statistical based tools to assist the modeler in discovering, describing and following the evolution of groups of agents, by combining data clustering and value test. Our model can be described within 5 main steps which will be illustrated with a NetLogo library model example.

2. ANALYSIS MODEL

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012),* Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

2.1 Model selection : what do we study?

Our model is generic for a simulation data stream. It may be generated directly from a simulation framework (NetLogo in our application) or extracted from a log file (by simulating an online data stream). We choose here the Bank Reservs model, provided with Netlogo, where financial agents either save money or borrow money via loans. In this model, some global variables give an overview of an experiment: we can observe phenomena as the stabilization of the total *money* when the maximum amount of *loans* is reached, or follow fixed groups of agents that depend on global parameters (for instance 3 groups: negative *wealth*, *wealth* higher than $richThreshold$ and the rest). Even if these informations are interesting, a more precise understanding of the model behavior can not be reached with such global/local observation. For example, *who are the wealthy agents? Do the rich stay rich?* This would be even more true for more complex models, for which variable interactions are much more difficult to deduce than with such very simple toy simulation.

2.2 Data processing : what is the data?

A data matrix is generated every *st* steps. A line in the matrix represents one agent's state. These raw data are not the only interesting variables for cluster's generation and analysis. From the set V_{ag} of variables that describes the agents, we generate the $\mathbb{V}calc$ and $\mathbb{V}init$ sets. In the $\mathbb{V}calc$ set, several filters/agregators can be considered to enrich the data stream. For now, we use one: for each variable, we add a new variable computed as its moving average. The $\mathbb{V}init$ set contains the initial values of V_{ag} of every agent in the simulation. By default, this *initial* variables are not used in the clustering but used for the later cluster's description. The subset of variables used for the clustering $\mathbb{V}clust$ has to be selected at the beginning of the simulation. By default, $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$.

2.3 Clustering: can we find homogeneous groups?

Clustering is performed on the $\mathbb{V}clust$ data set in order to generate homogeneous agent groups. Our objective is not to propose a new clustering algorithm, and any clustering algorithm may be used (in our application, any algorithm of the Weka framework and the associated parameters). By default, the XMeans clustering algorithm is used with the classical similarity function based on the Euclidean distance.

2.4 Cluster's description: how can we describe them?

Once the clusters are identified, and given $\mathbb{A}$ the entire set of agents and $\mathbb{A}_C$ the set of agents in a cluster C it is possible to obtain easy-to-read descriptions of every cluster by using the value tests VT calculation (equation 1). The VT represents the significance of the average value $E(\mathbb{A}_C^v)$ of a given variable v for the agents in the $\mathbb{A}_C$ set compared to its distribution on the $\mathbb{A}$ set. Roughly, it can be described as the difference between the average of the Cluster ($E(\mathbb{A}_C^v)$) and the global average ($E(\mathbb{A}^v)$) normalized by the standard deviation ($\sigma^2(\mathbb{A}^v)$):

$$VT(v,C) = \frac{(E(\mathbb{A}_C^v) - E(\mathbb{A}^v))}{\sqrt{(\frac{sizeof(\mathbb{A}) - sizeof(\mathbb{A}_C)}{sizeof(\mathbb{A})-1} \times \frac{\sigma^2(\mathbb{A}^v)}{sizeof(\mathbb{A}_C)})}} \quad (1)$$

v is significant for the A_C set if $|VT(v,C)| \geq 2$. If $VT(v,C) \geq 2$, the value of v in A_C is in average higher than in A (and lower if $VT(v,C) \leq -2$).

A global overview of all clusters retrieved in a simulation makes it easy to compare clusters, to describe them and the most significant variables in the simulation. For example :

"who are the wealthy?" We use our analysis tool with the default configuration: $\mathbb{V}clust = \mathbb{V}ag \cup \mathbb{V}calc$. The clustering step is fixed to $st = 200$ simulation steps. At each clustering step, clusters are identified, visualized in NetLogo (with colors), and their extension and description are presented. For example, in $t = 400$, three clusters are identified: a "*poor*" cluster (114 agents, with low *wealth*, $VT = -9.57$), a "*medium*" cluster (20 agents) and a "*wealthy*" cluster: *cluster*9. This cluster is composed of the 66 rich people, with high *wealth* ($VT = 8.91$), *savings* ($VT = 9.68$) and few *loans* ($VT = -3.93$), the results for the associated moving average variables are similar. An interesting result is the significant $T0Wallet$ ($VT = 8.81$) variable, corresponding to the *wallet* value of agents at the beginning of the simulation. The *wealthy* people were significantly richer than the average at the beginning of the simulation.

"what are the rich characteristics at each step?" At the end of the simulation, a description of all the clusters obtained at each step gives a global overview of the simulation. In our experiment, it is always possible to identify a *wealthy* and a *poor* cluster, and sometimes (like in $t = 400$) a *middle* cluster. From their description, it is already possible to observe that the link between the *wealth* and the initial wealth (the $T0Wallet$) is not significant anymore after $t = 400$ ($|VT(T0Wallet, rich)| < 2$ for $t \geq 400$). It may be related to the fact (observed with NetLogo global observation) that bank has reached its loan limit (the total money stops to increase around $t = 230$). However, to compare clusters of different steps in this overview is difficult since they are different both in intension and in extension (see section 2.5). In a more complex model, cluster may have a completely different meaning at different steps.

2.5 Cluster's evolution: how do they evolve?

In order to describe the clusters' evolution, we consider two alternative hypothesis: either the extension or the intension in every cluster is considered as stable.

Fixed extension:*Do the rich stay rich?*

The first possibility is to fix the extension of a cluster (its population), at a given simulation step t. The description of the cluster C is updated at every step. Since the agents in the cluster stay the same, $VT(v_{init}, C_{t'}) = VT(v_{init}, C_t) \forall t' > t$ (unless some agents die or new agents enter the simulation).

If we consider for example the *wealthy* agent of *cluster*9, the initial parameters values ($\mathbb{V}init$) are stable (by definition). But for some other variables, all wealth-related, the differences with the other agents decrease: both *wealth*, *saving* and *loans* (absolute) VT values decrease. This mean that, in average, the wealthy people of $t = 400$ are less and less wealthy. Even if they are still significantly wealthier than the average in $t = 1200$, but their *loans* variable is already not significantly lower anymore from $t = 1000$.

Fixed intension:*Do the rich stay the same people?*

The alternative is to fix the intension of clusters. The intension of a set of clusters is the function that allows to assign an agent to the most relevant cluster. At each new step, the intension function is used to determine which agents to put in each cluster (some of them may be empty) and the new descriptions are computed. For a cluster C_t, for every timestep $t' > t$ a new cluster $C_{t'}$ is built. The description of every $C_{t'}$ can be easily compared with that of C_t. Since the cluster intensions are the same, the v-tests of the variables used in clustering ($\mathbb{V}clust$) are very likely to stay the same. But the v-tests of other variables, especially those of the initial variables ($\mathbb{V}init$) of the agents may evolve (since the population of the cluster changes).

In our example, for clusters of $t = 400$, all the variables considered in clustering ($\mathbb{V}clust$) are by definition roughly similar. However, the other variables evolve (in our example, the initial parameters of the agents $\mathbb{V}init$). *cluster*9 will for example regroup the wealthy agents at each step, but the number and the initial properties of its agents evolves. The number of wealthy agent ($card(cluster9, t)$) stay approximatively constant (66, 71, 56, 58, 65), but the evolution of the initial parameters confirms the observation made with the global overview: after $t = 600$ $|VT(TOWallet, rich)| < 2$ (the initial wealth of the agent ($TOWallet$) is not significant anymore after $t = 600$).

3. CONCLUSIONS

The simulation's observation framework that we present here, provides the modeller with generic tools that allow him/her to get a synthetic descriptive view of simulations. It can be used to understand the dynamics of simulations and to ease their validation. To allow the analysis of a wide number of different type of simulations we are currently adapting our framework to both consider qualitative and network variables and facilitate large simulations analysis. The latter will be done by integrating our framework to GAMA[4] and to the SimExplorer/OpenMole[3] engine.

4. REFERENCES

[1] F. Gaillard, Y. Kubera, P. Mathieu, and S. Picault. A reverse engineering form for multi agent systems. In *ESAW 2008*, pages 137–153, 2008.

[2] J. Gil-Quijano, T. Louail, and G. Hutzler. From biological to urban cells : lessons from three multilevel agent-based models. In *PRACSYS 2010*, Kolkota, India, 2010. LNCS.

[3] M. Leclaire and R. Reuillon. www.openmole.org.

[4] P. Taillandier, A. Drogoul, and D.-A. Vo. Gama: a simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In *PRIMA 2010*, 2010.