Modeling Social Influence in Social Networks with SOIL, a Python Agent-Based Social Simulator

Eduardo Merino, Jesús M. Sánchez, David García,
J. Fernando Sánchez-Rada, and Carlos A. Iglesias(✉)

Intelligent Systems Group, DIT, E.T.S. de Ingenieros de Telecomunicación,
Universidad Politécnica de Madrid, 28040 Madrid, Spain
{eduardo.merinom,jesusmanuel.sanchez.martinez,
david.garcia.martin}@alumnos.upm.es, {jfernando,cif}@dit.upm.es
http://www.gsi.dit.upm.es

Abstract. The application of Agent-based Social Simulation (ABSS) for modeling social networks requires specific facilities for modeling, simulation and visualization of network structures. Moreover, ABSS can benefit from interactive shell facilities that can assist the model development process. We have addressed these problems through the development of a tool called SOIL, which provides a Python ABSS specifically designed for social networks. In this paper we present how this tool is applied to simulate viral marketing processes in a social network, and to evaluate the model with real data.

Keywords: Social network · SOIL · Python · Viral marketing · Brand reputation · Rumor propagation

1 Introduction

Social networks have become relevant in our professional and personal relationships. Thus, social network analysis and simulation can be effective for understanding and exploiting homophily and social influence processes in social networks. Marketing techniques are usually applied to exploit social influence in social networks, in applications such as viral or word-of-mouth marketing, rumor spreading and online reputation management. This paper complements the demo presented at PAAMS 2017 on the use of the Python-based ABSS SOIL tool for social network modeling and analysis, which is illustrated with a number of developed models.

2 Main Purpose

SOIL aims at providing a research environment for ABSS in Python, with a strong focus on interoperability with existing libraries. It integrates with the

Y. Demazeau et al. (Eds.): PAAMS 2017, LNAI 10349, pp. 337–341, 2017.
DOI: 10.1007/978-3-319-59930-4_33

popular network processing library NetworkX[1] and with network visualization tools such as Gephi[2].

3 Demonstration

In this paper we present a case study that models the social influence of users in the social network Twitter. In particular, we study the role of social influence in rumor propagation and brand monitoring. In both applications, a diffusion message (rumor or brand advertisement) is propagated in the social network with the aim of infecting users. Users are considered infected when they accept or embrace the content of the message. The model presented is $M_{2.2}$ [4]. Twitter users are modeled as agents which can be in three states: *neutral*, if they are not affected by the message; *infected*, if they accept the message; *vaccinated*, if they have not been infected yet and believe in the antirumor or are infected by a message from a different brand; and *cured*, if they have been infected, but now believe the antirumor or are infected by a different brand. Additionally, the model includes a specific kind of users, called beacons, which detect the propagation of the message and try to combat it. Beacons are modeled after authorities that prevent rumor diffusion and competing influencers in social media. Agents include two additional states, beacon-off and beacon-on to represent beacons before and after detecting a rumor in a close node (neighbor).

The spread model starts with an initial number of infected users. In every simulation step, the state of each user may change though a series of interactions, each of which happens with a different probability. Infected users try to infect their neutral neighbors. Neutral agents may also become vaccinated with a given probability based on external factors (i.e. news). Vaccinated users attempt to cure or vaccinate their neighbors. Lastly, beacon agents spread anti-rumors to their neighbors, and follow these neighbors' contacts.

Table 1. Datasets of Twitter rumors and brand monitoring

Dataset	Number of tweets	Purpose	Period	Reference
Ford	348	Brand monitoring	13 months	[1]
Toyota	582	Brand monitoring	14 months	[1]
Obama	4975	Rumor propagation	8 days	[3]
Palin	4423	Rumor propagation	10 days	[3]

We have validated this diffusion model on four datasets (Table 1). The first two datasets (Ford and Toyota) are subsets of the Replab dataset [1], which focuses on monitoring the reputation of companies and individuals in Twitter.

[1] https://networkx.github.io/.
[2] https://gephi.org/.

Each tweet is classified as related (or unrelated) to an entity, the polarity for the entity's reputation (positive, negative or neutral), and the priority of the topic cluster the tweet belongs to (alert, midly important, unimportant). We have filtered the dataset and selected two automotive brands, Ford and Toyota, which can simulate how two brands advertise themselves on social media. In this case, the advertisement message is propagated and succeeds if the brand gets a good reputation. The last two datasets (Obama and Palin) are rumor datasets [3] that deal with identifying the spread of misinformation in social networks, such as Obama being a muslin or Palin's divorce. The dataset is labeled as endorses (propagate the rumor), denies (deny the rumor), questions (doubt about rumor credibility) or unrelated (not related to the rumor).

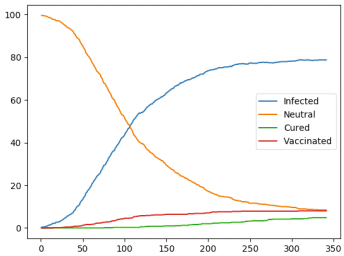

Fig. 1. Agent evolution

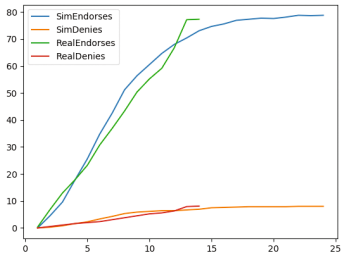

Fig. 2. Realism evaluation

The demonstration may be run in an IPython interactive shell, where simulation parameters can be defined. After running the simulation, the results are stored as Python objects, which can be inspected and visualized. For example, Fig. 1 shows the temporal evolution of agent states. The x axis represents the days and the y axis the number of simulated agents. In addition, the platform includes facilities for evaluating the realism of the simulation. For this purpose, we compare the daily number of endorsers and deniers in the dataset and the simulation. Figure 2 shows a comparison for the dataset of Toyota as a monthly evolution of the ratio of users that accept the diffusion message (endorsers) or reject it (deniers).

In addition, the platform generates a Graph Exchange XML Format (GEXF) file that can be used for analyzing the simulation with network analysis tools such as Gehpi. In particular, the visualization can be animated to show the temporal evolution of the spread model. Figure 3 shows a screenshot of the animation, where the colors denote infected (red), vaccinated (blue), cured (green) and beacon-off (yellow). Another interesting experiment is validating the realism of the simulation. An alternate view of the network is shown in Fig. 4.

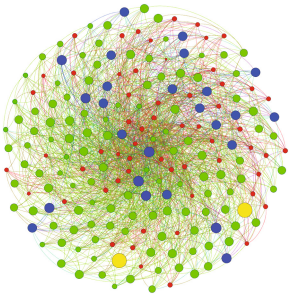

Fig. 3. Network visualization in Gephi (Color figure online)

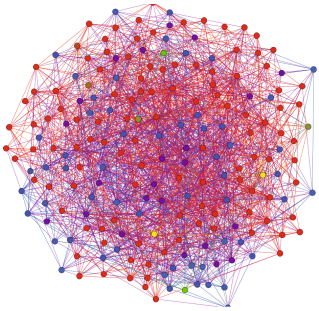

Fig. 4. Alternate network visualization

4 Conclusions

This demonstration shows the application of a Python ABSS specifically designed for social network modeling and its application to information diffusion in social networks. The models in this paper had an existing implementation written in Java [4], combining MASON [2] and the graph library GraphStream[3]. Porting them to SOIL was straightforward and resulted in much simpler comprehensible code. The main benefits from using SOIL derive from using a simple yet extensible interface and the Python programming language. As a result, it is very easy to extend agent behavior while leveraging the existing ecosystem to integrate machine learning algorithms or semantic interfaces, to name a few. Moreover, the use of an interactive shell such as IPython[4].

Acknowledgements. This work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D projects SEMOLA (TEC2015-68284-R) and EmoSpaces (RTC-2016–5053-7), by the Regional Government of Madrid through the project MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER), and by the European Union through the project MixedEmotions (Grant Agreement no: 141111). The authors want to thank Vahed Qazvinian for making available the rumor datasets for our research.

References

1. Amigó, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Martín, T., Meij, E., Rijke, M., Spina, D.: Overview of RepLab 2013: Evaluating online reputation monitoring systems. In: Forner, P., Müller, H., Paredes, R., Rosso, P., Stein, B. (eds.) CLEF 2013. LNCS, vol. 8138, pp. 333–352. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40802-1_31
2. Luke, S.: MASON: A multiagent simulation environment. Simulation **81**, 517–527 (2005)

[3] http://graphstream-project.org/.
[4] https://ipython.org/.

3. Qazvinian, V., Rosengren, E., Radev, D.R., Mei, Q.: Rumor has it: Identifying misinformation in microblogs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1589–1599. Association for Computational Linguistics (2011)
4. Serrano, E., Iglesias, C.A.: Validating viral marketing strategies in twitter via agent-based social simulation. Expert Syst. Appl. **50**(1), 140–150 (2016)