Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

PIPELINES AND THEIR COMPOSITIONS FOR MODELING AND ANALYSIS OF CONTROLLED ONLINE NETWORKED SOCIAL SCIENCE EXPERIMENTS

Vanessa Cedeno-Mieles

Biocomplexity Institute of Virginia Tech
Department of Computer Science
1015 Life Science Circle
Blacksburg, VA 24061, USA

ABSTRACT

There has been significant growth in online social science experiments in order to understand behavior at-scale, with finer-grained data collection. Considerable work is required to perform data analytics for custom experiments. We also seek to perform repeated networked experiments and modeling in an iterative loop. In this work, we design and build four composable and extensible automated software pipelines for (1) data analytics; (2) model property inference; (3) model/simulation; and (4) results analysis and comparisons between experimental data and model predictions. To reason about experiments and models, we design a formal data model. Our data model is for scenarios where subjects can repeat actions (from a set) any number of times over the game duration. Because the types of interactions and action sets are flexible, this class of experiments is large. Two case studies, on collective identity and complex contagion, illustrate use of the system.

1 INTRODUCTION

Online controlled networked social experiments (games) are increasingly used to study social behaviors and explore phenomena such as collective identity, exploration versus exploitation, and diffusion and contagion. Computational modeling is useful in understanding and reasoning about these behaviors. Combining experiments and modeling enables each to inform and guide the other. An iterative experimental and modeling approach requires several classes of operations: (1) design and conduct experiments, (2) data acquisition, (3) data fusion and integration, (4) analyze experimental data, (5) develop and verify models, (6) infer model parameters, (7) run simulations, (8) compare experimental data against model output, (9) exercise models beyond the ranges of experimental data, and (10) iterate. To improve human productivity, it is advantageous to automate these operations for improved efficiency, reproducibility, and scalability. Although there are software systems that address some of these operations, an automated and extensible system for evaluating social phenomena through iterative experiments and modeling that addresses all of these issues is lacking.

2 NOVELTY AND CONTRIBUTIONS

The novelty of this work is devising a data model and applying a computational model that together form abstract representations of experiments and modeling and simulation (MAS) so that we can determine whether an experiment or simulation can be analyzed with our system, and ensure correspondence between experiments and MAS. A second novelty is that our pipelines take a microservices conceptual approach wherein the software components of a pipeline, which we call functions, have narrow scopes. A third novelty is the range of capabilities we currently provide: most workflows in the social sciences are confined to static social network analyses; we go well beyond that here, including dynamics on networks.

We develop a formal abstract data model for networked social science experiments. The model provides a common representation for both experiments and modeling, thus producing a correspondence between experiments and MAS. We design and construct the following pipelines (1) Data Analysis Pipeline (DAP), (2) Property Inference Pipeline (PIP), (3) Modeling and Simulation Pipeline (MASP), and (4) Model Evaluation and Prediction Pipeline (MEAPP). Two case studies, on collective identity and complex contagion, illustrate use of the system: (1) shows a full system execution for collective identity experiments; and (2) represents the effects of network structure on complex contagion diffusion (Cedeno-Mieles et al. 2018).

3 DATA MODEL AND GRAPH DYNAMICAL SYSTEM MODEL FOR NETWORKED EXPERIMENTS AND MODELING AND SIMULATION

Data Model. Each experiment has a unique id *exp_id*, a number n_p of phases, a number n of players, a *t_begin* timestamp for the beginning of the game, a *t_end* timestamp for the end of the game. Each player has a unique id v_i for identification. A set of players in an experiment is defined by V = $\{v_1,.., v_n\}$. An experiment has n_{sa} player attributes defined for each player. Player attributes Ω are invariant across phases. Each phase schema has the following elements, a unique id *ph_sch_id*, the number i_{n_p} of the phase in the sequence of phases, a *t_ph_begin* timestamp at the beginning of the phase, number t_p of time increments in the phase and the unit of time u_p of one time increment. Each phase represents the interaction structure among players as a network *G(V',E')* with meanings of edges Λ. Node attributes Γ and edge attributes Ψ over all nodes and edges capture attribute changes in time. Players and edges may have initial conditions B^v and B^e, respectively. A is the set of permissible player actions. An action tuple T_i, which captures pairwise interactions between players, may be intimately tied to the attribute sequences Γ and Ψ of a phase because action tuples, for example, may cause or be caused by changes in node and edge attributes.

Graph Dynamical System (GDS). We use GDS to specify, build, and execute experiments and simulators of experiments. We use this GDS model because it is correspondent with the presented data model and because GDSs represent a general model of computation since they can simulate Turing Machines. A synchronous GDS *S* is specified as *S = (G, F, W),* where (a) *G(V, E),* an undirected graph with $|V| = n$, represents the underlying graph of the GDS, with node set *V* and edge set *E*, (b) $F = (f_1, f_2,.., f_n)$ is a collection of functions in the system, with f_i denoting the **local function** associated with node v_i, $1 \leq i \leq n$, and (c) W is the state space, which is the union of the state space W^v for nodes and the state space W^e for edges; i.e., W = $W^v \cup W^e$. Each undirected edge $\{v_i,v_j\} \in E$ can be represented by two directed edges: v_i to v_j, (v_i,v_j), and (v_j,v_i).

4 HIERARCHICAL PIPELINE CONCEPTUAL VIEW AND IMPLEMENTATION

Pipelines. (1) The DAP analyzes temporal interactions among players to identify patterns and phenomena in the data and serves as the input for (2), the PIP. The simulation models (e.g., agent-based models, ABMs) are built off-line and are part of (3) the MASP. (4) The MEAPP combines simulation results across multiple (stochastic) executions and performs comparisons between experimental data and model predictions.

REFERENCES

Cedeno-Mieles, V., Y. Re, Z. Hu, X. Deng, S. Ekanayake, B. J. Goode, C. J. Kuhlman, D. Machi, M. V. Marathe, H. H. Mortveit, N. Ramakrishnan, P. Saraf, N. Self, N. Contractor, J. M. Epstein, and M. W. Macy 2018. "Pipelines and their Compositions for Modeling and Analysis of Controlled Online Networked Social Science experiments". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe et al. *To Appear*. Piscataway, New Jersey: IEEE.