Sensitivity Analysis of an Agent-Based Simulation Model Using Reconstructability Analysis

Andey M. Nunes
Portland State University, andey@pdx.edu

Martin Zwick
Portland State University, zwick@pdx.edu

Wayne Wakeland
Portland State University, wakeland@pdx.edu

Citation Details

Published as: Andey M. Nunes , Martin Zwick & Wayne Wakeland (2021): Sensitivity analysis of an agent-based simulation model using reconstructability analysis, International Journal of General Systems.

Sensitivity Analysis of an Agent-Based Simulation Model using Reconstructability Analysis

Andey M. Nunes, MS (corresponding author)
Systems Science Program, Portland State University, Portland, OR 97207
503-869-1300 andey@pdx.edu

Martin Zwick, PhD
Systems Science Program, Portland State University, Portland, OR 97207
503-725-4987 zwick@pdx.edu

Wayne Wakeland, PhD
Systems Science Program, Portland State University, Portland, OR 97207
503-725-4975 wakeland@pdx.edu

Author Bios

Andey Nunes is a research analyst at Oregon Health Authority - Public Health Division. He earned his MS – Systems Science at Portland State University in 2017 and BS – Environmental Science, Earth Systems Science from San Francisco State University in 2012. His current pursuits focus on applied statistical analytics and decision support systems in Emergency Medical Services and Trauma Systems.

Martin Zwick is a Professor of Systems Science at Portland State University. Prior to taking his current position at PSU, he was a faculty member in the Department of Biophysics and Theoretical Biology at the University of Chicago, where he worked in macromolecular structure and mathematical crystallography. In the 1970's his interests shifted to systems theory and methodology. Since 1976 he has been on the faculty of the PSU Systems Science Ph.D. Program and during the years 1984-1989 he was the program head. In 2009, he was awarded the

Branford Price Millar award for faculty excellence His current research interests are in discrete multivariate modelling (reconstructability analysis), theoretical biology, and systems theory and philosophy.

Wayne Wakeland is the Program Chair of the Systems Science Graduate Program at Portland State University. He earned his Ph.D. in Systems Science from Portland State University in 1977 and began teaching modelling and simulation in 1978 as an adjunct member of core faculty in the Systems Science Ph.D. program. While teaching part-time from 1978 to 2000 he held managerial positions in information systems, technology, and manufacturing companies. He became Associate Professor of in 2000, and Professor in 2016 in the Systems Science Program and was awarded the 2017 George C. Hoffman award for faculty excellence. His research interests include: ecological economics and sustainability, health systems policy, biomedical dynamics, the software development process, criminal justice systems, supply chain management, organizational dynamics and systems thinking, simulation & optimization methods, and teaching.

Word Count

Manuscript is 6644 words (not including reference and placeholder text for tables and figures).

Abstract

Reconstructability analysis, a methodology based on information theory and graph theory, was used to perform a sensitivity analysis of an agent-based model. The NetLogo Behavior Space feature was employed to do a full 2^k factorial parameter sweep on Uri Wilensky's Wealth Distribution NetLogo model, to which a Gini-coefficient convergence condition was added. The analysis identified the most influential predictors (parameters and their interactions) of the Gini-coefficient wealth inequality outcome. Implications of this type of analysis for building and testing agent-based simulation models are discussed.

Key words: reconstructability analysis; machine learning; agent-based simulation; information theory; sensitivity analysis; wealth distribution model

1. Introduction

In agent-based simulation (ABS), agents interact with each other in a dynamic environment. By following simple rules, these interactions result in emergent behavior patterns. SugarScape is a widely studied ABS model developed by Joshua M. Epstein and Robert Axtell (Epstein and Axtell 1996). The NetLogo Wealth Distribution model, developed by Uri Wilensky, is based on the SugarScape model and includes output variables for the Gini coefficient, a measure of wealth inequality, and for the class distribution in the simulation population. This project applied a machine learning methodology to the outputs generated by Wilensky's Wealth Distribution model to answer the following questions, *"Can a machine learning algorithm detect relations between model parameters and model output that augment our understanding of the model? Specifically, can such an algorithm reveal the degree to which the model parameters and their interactions predict the model output?"*

To address these questions, data produced by simulations of the NetLogo Wilensky Wealth Distribution (WWD) model were analyzed with a software tool called OCCAM (named after the principle of parsimony – or 'Organizational Complexity Computation and Modeling'). OCCAM implements a machine learning methodology known as Reconstructability Analysis (RA), well suited for detecting nonlinear and high ordinality multivariate interactions, and is available both online and as open-source code (Zwick 2019). The results of the OCCAM analysis illuminated the sensitivity of Gini coefficient outcomes to the parameters and interactions between the parameters in the model.

RA modeling of data from WWD simulations is a particularly powerful type of sensitivity analysis of the WWD model, since sensitivity analysis usually involves varying input parameters one at a time to see how variation affects the output, i.e., it is commonly done without considering interactions among the inputs. Our sensitivity analysis is much more substantial, since we do not look only at main effects. We deploy a full-scale RA analysis on top of the WWD simulation, i.e., we add data-driven modeling (RA), as a meta-level, to theory-driven modeling (WWD), as the base level. In theory-driven modeling one posits a set of theoretically plausible relations between variables. In the data-driven modeling, by contrast, relations are derived directly from data rather than from theory. For the WWD model, one has a theory-based expectation of a property that will emerge from the hypothesized relations, namely income inequality, but one does not have theoretical expectations about how this property will actually depend upon the model parameters. To discover this dependence, we applied RA to data generated by WWD simulations. RA is a general machine learning methodology which could be applied to data from any simulation, but it is likely that other machine learning approaches would also usefully supplement agent-based simulation. Our purpose here was not to advocate specifically for RA., and we have not compared its effectiveness to other machine learning approaches. Our aim is primarily to offer a proof of concept: to show that adding a machine learning post-processing step usefully augments ABS. We expect that our proof of concept will suggest new modeling possibilities to researchers, since such two level analyses are rare in literatures of both simulation and machine learning. Demonstrating the capabilities of RA, the specific methodology that we used, is only a secondary aim of this paper. However, since RA is less well known than other machine learning methodologies, this demonstration adds value to this study.

The Wealth Distribution NetLogo Model

Economists Joshua M. Epstein and Robert Axtell's 1996 book *Growing Artificial Societies*, introduced the SugarScape model and the idea of using agent-based simulation as a form of generative social science research. Overall patterns of population behaviors emerge from the simple rules involving individual agents' fitness parameters, the abundance or scarcity of resources in the environment, and population dynamics (Epstein 1999, Wilensky and Rand 2015). The inspiration for studying the SugarScape model and the focus of this paper is best articulated in the following quote from Epstein & Axtell (1996) regarding the importance of studying agent-based models: "The ability to alter agent-interaction rules and compute the effect on the Gini-coefficient and other summary statistics is one of the most powerful features." The aim of this study is to explore to what degree a machine-learning algorithm can predict a macro-emergent condition – the Gini coefficient – from the simulation parameters.

Based on SugarScape, Wilensky's Wealth Distribution model is included in the NetLogo models library with the two additional output variables: the Gini-coefficient and a class histogram (for low, middle, and upper class) to display the overall distribution of the primary resource among the simulated population of agents. The primary resource is sugar in SugarScape and is grain in the Wealth Distribution model. The outcomes of both models demonstrate the Pareto Principle with most people being poor, some middle-class, and a very few being wealthy; and the richest 20% of the population hold 80% (or more) of the total wealth (Wilensky 1998).

The following description of the Wealth Distribution model is summarized directly from Wilensky's (1996) **Info** tab in the NetLogo model library. When the model is set up, the environment, consisting of equally sized *patches* (nonmobile agents) in a two-dimensional plane, is endowed with a random assignment of grain and grain growing capacity. A population of individuals is randomly endowed with an initial wealth level and fitness characteristics, and then randomly dispersed throughout the environment. The model is executed in time-steps, where at each step, individuals look around at neighboring patches for grain, move towards the most plentiful patch within the limits of their visual capabilities, and harvest. Each time step involves this maneuver and costs the individual the amount of grain specified by their random metabolism assignment. After harvest, patches re-grow grain according to their random assignment for growth patterns. The calculation of the wealth distribution for individuals and population is executed and updated in the NetLogo interface. An individual agent that fails to find enough grain to meet its metabolism demand does not survive and is replaced with another randomly generated individual in order to maintain the population number. Agents can also expire by meeting the limit of their randomly assigned life expectancy. At each step the wealth distribution is determined by ranking the individuals according to portion of total population of wealth owned and then calculating the Gini coefficient.

When Wilensky, Epstein, Axtell, Resnick and others wrote about agent-based simulation models they discuss agents as being anything in the model that can be coded to follow simple rules. In NetLogo, agents are then distinguished between environmental agents, which they called patches and the individuals which they called turtles. In this paper, we will use the term *agents* in a more narrowly defined way to refer only to the population of individuals.

The following table summarizes the parameters in the Wealth Distribution model.

(Table 1)

Reconstructability Analysis and the OCCAM software tool

Only a brief description of Reconstructability Analysis (RA) is provided here. Cornerstones of RA include Weiner's (1914) work in set theory and relations, Shannon's (1948) concept of entropy, Ashby's (1964) constraint analysis, and Klir's (1969) general systems theory (Jones 1985, Klir 1985, Zwick 2004). Foundations of RA are information theory, graph theory, classical set theory, and probability theory (Klir 1986, Zwick 2004). RA overlaps with log-linear methods, Bayesian networks, and other probabilistic graphical modeling methodologies and is applicable to both nominal and continuous multivariate data (Zwick 2004). It is qualitatively different from continuous variable methods such as neural networks and regression techniques.

Klir defined RA as a methodology that deals with the class of problems characterized by the relationship between an overall system, referred to here as the whole, and the multiple subsystems – mathematically, the projected relations – that comprise the structure of the system, referred to here as the parts (Klir 1985, Zwick 2001). The aim of the most standard uses of RA is to find the simplest set of parts from which a good approximation to the whole can be constructed (Klir 1985, Zwick 2004). The whole is an observed relation in data; the approximation to the whole from a set of parts is a calculated relation. The synthesis of the calculated relation is done using a maximum entropy formalism, which typically gives results equivalent to maximum likelihood calculations.

The two versions of RA are information-theoretic, which applies to frequency and probability distributions, and set-theoretic, which applies to set-theoretic relations and mappings (Klir 1985, Krippendorff 1986, Zwick 2004). Both versions use the same Lattice of Structures for the exploration of possible models. However, set-theoretic RA utilizes Hartley entropy, is non-statistical, and overlaps with logic design and machine learning methodologies (Zwick 2004), while information-theoretic RA uses Shannon entropy and a Chi-squared distribution to assess models for statistical significance, similar to log-linear methods (Knoke & Burke 1980, Krippendorff 1986, Zwick 2004). This study uses information-theoretic RA to predict the Gini outcomes from the model parameters of the NetLogo agent-based model.

OCCAM is a web-based RA software package for exploratory modeling (Zwick 2019). It can be used to analyze data sets involving nominal variables or continuous variables that are binned (discretized). It performs a beam search of the Lattice of Structures and summarizes this search with the three best models based on Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC), and a third criterion that seeks the highest information model that is ‘cumulatively’ statistically significant relative to independence and ‘incrementally’ significant for every step in some path from independence to the model. These statistical tests use the Chi-square distribution and a user-specified p-value cutoff where 0.05 is the default value. Usually, a model is selected using one of these three criteria, and the model’s conditional distribution of the dependent variable (DV) – here the Gini coefficient – given the independent variables (IV) – here the ABS parameter settings – is used to predict the DV. In this study, models were first fit on training data, and then applied to test data. The goodness of a model’s prediction is quantified by the model’s reduction of uncertainty of the DV, given the model’s predicting IVs, and the percent correct (%C) in the test predictions.

Three types of models were considered – models without loops, disjoint models, and all models (including those with loops) – and thus three model searches were performed. These model types can be illustrated as follows. Suppose one has three IVs, namely A, B, and C, and one DV, namely Z. A model without loops has the form IV:ABZ where the ‘IV’ relation in the model means a relation involving all the IVs, which here is ABC. The ABZ relation in this model says that there is an interaction effect between IVs A and B with the DV, Z. Models without loops pick out a single subset of predictors from among the IVs. In other context, this search is useful for feature selection, but in this study, all IVs are retained in all searches, so loopless models are of interest only for their simplicity. The results of a loopless model search where the predicting relation involves only one IV is given ahead in Table 4, which lists the individual IVs in order of predictive strength.

An example of a model *with* loops is IV:ABZ:BCZ. This model has the usual relation among all the IVs, plus two predicting relations, ABZ and BCZ; each of these predicting relations involves a three-way interaction effect. Such models are invariably more predictive of a DV than loopless models, but they require iteration to be fitted. A disjoint model is a simple type of a model with loops. It can have multiple predicting relations, but these relations are disjoint in the predicting IVs. An example is IV:ABZ:CZ, in which no IV is present in both predicting relations. Disjoint models also require iteration to be fitted.

An all-models search allows loops and overlaps of IVs in the predicting relations; it performs the best, i.e. it finds the most predictive models. A search restricted to disjoint models finds somewhat less predictive models, but the separation of predicting IVs into disjoint groups allows for simpler interpretation of the model. Finally, searches of loopless models find even less predictive models, but these models are maximally simple and easier to understand.

Models are fit on training data and their generalizability is assessed on test data. The predictive strength of a model on the training data is indicated by its reduction of uncertainty, its

ΔBIC value, and its percent correct on the training set (%Cdata). Its generalizing performance is assessed by percent correct on the test set (%Ctest). The statistical significance of the model relative to reference of independence, is given by a p-value ('alpha'). The complexity of a model, relative to independence, is its Δ*degrees of freedom*. These measures are summarized in Table 2.

(Table 2)

2. Methodology

The experiments conducted in this study using the NetLogo (v.5.3.1) Wealth Distribution Model represent a type of analysis that is similar to sensitivity analysis but more comprehensive. Wilensky and Rand describe the sensitivity analysis of an agent-based model within the context of model verification, validation, and replication (2015). The inquiry begins with the question, "Sensitive to what?", and depends on whether the results being considered are qualitative or quantitative (Wilensky and Rand 2015). Here we are interested in the outcome of the Gini coefficient when the value tends to converge, so a stopping criteria was added to the end of the code which tells the simulation to stop when the difference between the last step and the mean Gini-Index value for the last 25 steps is less than 0.001 or one-tenth of a percent. We used the NetLogo Behavior Space tool to run the simulation over selected variable settings and collect data on the Gini-Index at the end of the run. This section describes the process in two phases: the NetLogo simulation data collection and processing, and the OCCAM (v.3.3.11) simulation analysis.

Data collection and processing

The data collection approach for this experiment was adapted from Uri Wilensky and William Rand's *An Introduction to Agent-Based Modeling*, specifically it used NetLogo's BehaviorSpace tool to run the set of experiments to generate the large data set needed for the data mining application (Wilensky and Rand 2015).

NetLogo Wealth Distribution Model Parameter Definitions and Variable Descriptions

Of the NetLogo Wealth Distribution model variable names and definitions given in Table 1, the first five population parameters listed are *agent variables* which determine behavior and interactions of free-roaming agents, while the last four are *environmental variables* that determine how patches behave and interact. This set-up gives nine model parameters as IV predictors of the Gini-coefficient DV.

Both the time-step and the Gini coefficient are continuous variables, and must be binned or recoded into discrete categories before passing the data file to OCCAM for analysis. An Excel Macro tool designed for rebinning continuous data and formatting an OCCAM input file was used to recode the time-step into three bins, and the Gini coefficient into four equal interval bins where 1 corresponds to low values and 4 corresponds to high values of the Gini-Index outcome [1].

NetLogo BehaviorSpace experimental design and data collection

[1] The Gini coefficient recoded is interpreted as lower values representing more equitable distributions of wealth and higher values representing greater inequity among the population.

A full factorial 2^k statistical design was chosen to sample the model parameter space. Table 3 defines the parameter range and the experimental values chosen for the low and high states. The max-grain variable that is imbedded in the model code was given the baseline "as-is" model value of 50 as the low value, and a resource rich environment value of 500 for the high value. The simulation was done by setting the max-grain value in the code and running a full 2^9 design twice. This whole experiment was replicated twenty times for the training data and five times for the test data. A different random number generator seed was used for each replication.

(Table 3)

For each run, BehaviorSpace was set to record the ending time-step of the run and the converged Gini-coefficient value. Since the stopping condition used a running average of 25 time-steps for the Gini-coefficient, the earliest step the run would stop is 26 steps. Stopping time is not a model parameter or variable, but is used in this analysis as an IV to capture a possible relationship between the Gini-coefficient outcome and the number of steps before the equilibrium stopping condition is reached. This experimental design resulted in a state space size of 2^9 (for the first nine parameters in Table 3) x 3 (number of bins for the last Table 3 variable) x 4 (number of bins for the DV) = 6144. The 20 training replicates and the 5 test replicates gave a total sample size of 10240 runs for training data and 2560 runs for test data.

OCCAM analysis set-up

OCCAM analysis consists of two steps: *search* and *fit*. The search step was conducted using the default OCCAM settings as follows: default search direction *up*; sort by ΔBIC (dBIC) during search; when searching look for larger ΔBIC values, use alpha threshold of 0.05; sort report by descending information percentage, and include all reporting options for statistical calculations. A search width of 10 was selected for all searches for several reasons: to obtain the order in which single predictor models (for each variable) reduce uncertainty; to observe top 10 predictors at each search level; and to try to avoid best model summary results that are search path dependent. The number of levels searched depended upon the search type: for loopless models, 12 levels (one more than the number of variables); disjoint models searched 20 levels; and all models search was set to 70 levels. These values were selected experimentally as to have conducted each search to either the top of the lattice, or high enough that the three best models selected were one or more levels under the top-most searched lattice level.

The OCCAM search output provides a log of the report settings, a summary with selected statistical measures for top models at each search level, followed by a list of the *Best Models* by ΔBIC, ΔAIC, and Incremental-p-value. The last of these criteria picks the highest information model whose difference from the reference is statistically significant and for which a path from the reference to the model exists where each incremental increase in complexity is also statistically significant (for some user-specified p-value cutoff).

Since the input file contains both a training set and a test set of data, this list also includes a *Best Model by %C(Test)* with the warning that models should not be selected based on the percent correct in the test data. This fourth 'best model' just allows the user to see how close the three model selection criteria are to what *would have been* an optimal model for the test data if the DV values for the test had been known (which, for true test data is never the case). The *Best Model by %C(Test)* thus indicates for each of the three selection criteria whether it overfits or underfits the training data. These indications can be seen in Figures 1, 2, and 3.

The Fit step uses the ΔBIC best model from the *all model* search on the training data. This step displays the conditional probability distributions for this model on the training and test data, as well as the percent correct on training and test data for the model as a whole and for each relation in the model. It also shows for the model as a whole and for each relation how much their percent correct improves upon the reference model percent correct. Specifically, improvement varies from 0 to 1 and is given by

$$Improvement = \frac{\%C(model) - \%C(reference)}{\%C(highestpossible) - \%C(reference)}$$

where %C(highest possible) is the percent correct for rules that would optimally predict the DV given the IV states, which is not 100% because the data is stochastic. For some IV states, the DV outcomes have some probability distribution, so perfect prediction is inherently impossible. The optimal rule set predicts, for each IV state, the most probable DV state.

3. Results & Discussion

Search Results

We begin our analysis by using a loopless search to order each independent variable (IV) as a single predictor of the dependent variable (DV). By selecting the search width equal to the number of IVs, we can see the reduction of uncertainty in predicting the DV and the corresponding percent correct in predicting the training data and the test data, as shown in Table 4. This orders the single IV predictors by their strength in reducing uncertainty; the strongest predictors are likely to show up in the search log for the disjoint and all-models searches. Specifically, in this case the top five predictors consistently show up in the best models at each of the first several levels in every search mode that follows. For Table 4, and also Table 5 (ahead), the asterisk next to an ID number indicates that the model has satisfied the IncrP requirement.

(Table 4)

The minimum age of the agents is the poorest single predictor of the Gini coefficient as it has a negative ΔBIC value, which means that its BIC scores was worse than the reference model. The population variable was the best single predictor, followed closely by the vision variable. The table also shows how uncertainty reduction and the percent correct predictions for training and test data are not linearly related measures of model fitness. We see from these results that *Pop* is the only single predictor that improved the percent correct on both training data and test data over the independence model.

(Table 4)

Table 5 gives the best models summaries from each search type: loopless, disjoint, and all-models. Similar to Table 4, the asterisk (*) next to the model ID indicates that the model satisfies the IncrP requirement. For each of the search types that follows, details from the search log summary showing models from each level in the lattice are plotted to show the trade off between complexity as ΔDegrees of Freedom (ΔDF) and the reduction of uncertainty obtained by the model as the search moves up the lattice from the reference model. Figure 1 shows first the loopless model search lattice with 282 total models. Figure 2 shows the disjoint model search lattice with 1285 total models, and finally Figure 3 shows the all-model search lattice with 15429 total models.

(Figure 1)

(Figure 2)

(Figure 3)

(Table 5)

In the loopless search, IVs appeared in the order shown in Table 4. The ΔAIC model and the %C(test) model were the same model (shown also in Table 5 as model ID 87) yielding 52.85% reduction of uncertainty with 765 degrees of freedom. Defining a model that does worse than the %C(test) model and is more complex or less complex than this model as "overfit" or "underfit", respectively, the IncrP model overfit and the ΔBIC model underfit. The ΔBIC model was more conservative by not including the Pbland term that showed up in the ΔAIC and %C(test) model.

In the disjoint search, all three best models overfit the test data with the ΔBIC model being the closest to %C(test). All three best models from this search contained interaction terms with five or more variables, whereas the %C(test) model was much simpler with only a four-way, a three-way, and two single term interactions with the DV. For the all-models search ΔAIC overfit, but IncrP and ΔBIC both came very close to the %C(test) model with ΔBIC being slightly closer than IncrP.

In all OCCAM search results in Table 5, the best models include relations where interaction effects involve at least 2 IVs and in several relations, 3 or 4 IVs are present; this illustrates the capacity of RA to detect complex interaction effects. Hypergraph representations of the all-models search result best model by ΔBIC are included in the Supplemental Figures, after the References section.

Uncertainty reduction can be partitioned between the agent variables, environmental variables, and interactions between the two types of variables by providing only the variables of interest in the input file (by telling OCCAM to ignore the other variables in the data). Table 6 shows the results of this partitioning exercise. When all ten IVs are used the uncertainty reduction is 48.2%, whereas the agent variables provide 21.0% uncertainty reduction while the environmental variables provide 8.3% uncertainty reduction. The Time variable alone provides a fraction of one percent uncertainty reduction. What this shows is that while the agent variables have a considerably larger impact on uncertainty reduction than the environmental variables, there indeed is a substantial agent-environment interaction effect in reducing uncertainty for the Gini outcome. This is described in more specific detail in the Fit Analysis Results section that follows.

(Table 6)

Fit Analysis Results

The best model by ΔBIC from the all-models search (see Table 5) is:

IV: PopMetabVisGini**:** PopMetabGrrateGini**:** PopMetabPblandMaxgrGini**:** PopVisMaxageGini**:** PopVisGrintGrrateGini**:** PopVisPblandGini**:** PopVisMaxgrGini**:** PopGrratePblandGini**:** PopGrrateMaxgrGini**:** MetabVisGrintGini**:** MetabVisGrrateGini**:** MetabVisPblandGini**:** MetabGrintGrrateGini**:** MaxageMaxgrGini**:** GrintPblandGini**:** TimeGini.

The OCCAM Fit results summarize (for each relation) the frequencies for each IV state, the calculated conditional probabilities for the model, and the selected prediction rule. Additionally, the prediction rule gives the expected DV state along with percent and number correct on the data and the associated p-values. The Fit summary for test data includes a frequency table and percent correct based on the training data prediction rule, as well as a summary of the relation's performance on test data with the percent improvement by model based on the optimal prediction rule case for the test data. The OCCAM Fit analysis thus identifies which relations (which interaction effects) are the most important.

The best model in terms of ΔBIC from the all-models search contains the IV relation and sixteen model predicting relations. These predicting relations are listed in Table 7, starting on the second line, in order of percent correct on test data, %C(test). The first line of the table shows, for comparison, the percent correct of a model that is the data, namely that includes all ten IVs. Using all the IVs allows us to correctly predict the test data DV only 60% of the time; that is, the IV-DV relationship in the data is stochastic, not deterministic. Note that within the 16 relations, there are only two relations where all IV predictors are the agent variables and only one relation where they are all environment parameters; all the other thirteen relations involve interaction effects between one or more environmental and one or more agent parameters. All of the deviations of the conditional DV probability distribution given the composite IV states from both a uniform DV distribution and the marginal DV probabilities are statistically significant at the 0.00 level. Table 7 summarizes, for each relation, its percent correct and its percent improvement[2] (derived from equation shown on page 8) from the independence model baseline. A high %Improvement means that the prediction rule of the relation gotten from the training data comes close to the best possible prediction rule, namely the rule that would have been optimal for the test data. %Improvement does not have a simple relationship with %C(test) for the following reason. %C(test) is for prediction rules applied to the test data, where these rules are obtained from the probability distribution of the relation fitted to the training data, while %C(highest possible) is for prediction rules applied to the test data obtained from the probability distribution of the relation fitted to the test data. A relation, fitted as it should be on the training data, might predict very well, but nowhere near as well as if it had been fitted – illegitimately – on the test data itself.

(Table 7)

The relations are also summarized in Table 8 by frequency of predictors, and the last column gives the number of variables in each relation (listed in same order as Table 7) and the bottom row gives the number of relations containing each predictor variable (listed in order of single predictor strength, Table 4). The top two single predictors, namely *Pop* and *Vis* show up as the most frequent predictors in the all-models ΔBIC model relations. While the metabolism variable, *Metab,* was not in the top 5 of single predictor models, it did show up as the third most frequently occurring variable in the model relations which reveals its importance in terms of interaction effects.

(Table 8)

[2] $Improvement = \frac{\%C(component) - \%C(reference)}{\%C(highestpossible) - \%C(reference)}$

Additionally, the OCCAM Fit output provides, for each relation, the expected DV state for the various IV composite states. Here we will only discuss the Fit output of one of the 16 relations in the ΔBIC model, which is summarized in Table 9 and illustrated in Figure 4. The most predictive (highest %correct in test data) relation, PopVis*Maxgr*, is a three way interaction between the population size, the potential vision range for the agents, and the environmental max grain parameter. OCCAM provides for all relations, the conditional probability of all Gini states for each IV state. Table 9 shows part of this output. The first three columns specify the IV state. The "*Gini=*" columns are percentages of the frequencies for each IV state with these outcomes, where Gini=1 is the most equitable state and Gini=4 is the most inequitable state. The rule is a prediction rule based on the highest percentage DV outcome given the IV state; the probability of this DV state is shaded. Note that for two IVs states (the third and fourth rows of the table), the probability of the next lower Gini state is only slightly lower than the probability of the most likely Gini state; this is shown in the table with lighter shading, and the prediction rule indicates this alternative Gini state in parenthesis. Ratio_G1 and Ratio_G4 are the ratios of the probabilities of the predicted DV outcome for a given IV state relative to the marginal probabilities of Gini=1 and Gini=4. Ratio > 1.0 means increased probability of occurrence and Ratio < 1.0 mean decreased probability of occurrence. Red is used to indicate a tendency towards inequity and blue is used to indicate a tendency toward more equitable Gini states. Extreme cases are bolded. For each IV state, the conditional probability for the most probable DV state – which yields the prediction rule -- is shaded; states with probabilities close to the rule state are shaded more lightly. The frequency of every IV state is 1280; the total sample size is thus 10240.

(Table 9)

For example, the IV state (*Pop, Vis, Maxgr*) = (100, 15, 50) has rule = 2, which means that Gini=2 is predicted for this state, although Gini=1 is only slightly less probable. For this IV state, the probability of Gini=1 is .3383, 2.04 times the marginal p(Gini=1) = .1656, the probability of a highly equitable outcome. Also, for this IV state, the probability of Gini=4 is .0773, 0.49 times less likely than the marginal p(Gini=4) = .1584, the probability of a highly inequitable outcome. The table also shows that the IV state (1000, 15, 500) is neither more nor less likely to produce the most equitable outcome (Ratio_G1 = 1), but is extremely unlikely (Ratio_G4 = .02) to produce a highly inequitable outcome.

Here we can see that when the population is small (100) and vision is limited (1), and the environment is resource rich (500) the most inequitable Gini outcome is 4.24 times as likely to occur. Conversely, with a small population (100) with greater range of vision (15) and the baseline resource environment (50), the most equitable Gini=1 outcome is twice (2.04) as likely to occur, although the prediction rule is for Gini=2. Overall we see an increased probability of a more equitable Gini outcome (all Ratio_G1 values ≥ 1) when the population has a higher vision (Vis = 15); the values are 2.04, 1.62, 1.46, and 1.00. The prediction rules of the table can be summarized in a decision tree diagram in Figure 4.

(Figure 4)

4. Conclusions

Summary of Findings

The Introduction of this paper posed the following questions *“Can a machine learning algorithm detect relations between the model parameters and the model output that augment our* *understanding of the model? Specifically, can such an algorithm reveal the degree to which the parameters and their interactions predict the model output?"* We have shown that the answers to these questions are "*Yes. RA tells us how predictive are (i) single parameters, (ii) multi-parameter relations, and (iii) multi-relation models.*" The following summarizes the RA findings:

1) At its simplest (main effects) level, RA quantifies the degree of dependence of the model output upon each of the ten individual parameters in the model. It does so using an information theoretic measure (reduction of uncertainty) and allied measures (e.g., the Bayesian information criterion) that are more informative than more general %correct measure of prediction accuracy. The most predictive parameter, Pop, reduces the output uncertainty by 6.4% (Table 4). This may seem small, but %uncertainty reduction is very different from %variance explained because of the logarithm term in the expression for uncertainty; an uncertainty reduction of as little as 8% can correspond in some cases to the odds of two possible outcomes changing as much as from 1:1 to 2:1 (Zwick 2020). The second most predictive parameter, almost as predictive as the first, is the vision range parameter, Vis, which reduces the output uncertainty by 6.3%. These two parameters have considerably greater predictive power than the other eight parameters. There are five agent parameters, four environment parameters, and time, a neutral parameter. Both of the top two predictors are agent parameters.

2) The full RA analysis (Table 5) yields a best model that reduces the output uncertainty by 48.2%. Even though this model was selected using BIC, the most conservative model selection criterion available in OCCAM, the model is very complex, consisting of 16 predictive relations, which when fused together yield the 48.2% overall uncertainty reduction. 15 of these relations involve interaction effects where at least two parameters predict the output, while most relations

involve three predicting parameters (the 16th relation has time as a sole predictor). Of these 15 relations, 2 involve only agent parameters, 1 involves only environment parameters, while the remaining 12 involve *both* agent and environment parameters (Table 5, Table 7, Table 8). So the best BIC model is complex in the dependence of the output on the parameters in three different ways: (a) the model consists in multiple predictive relations, (b) nearly all these relations involve interaction effects of two or more parameters with the Gini output, and (c) most interaction effects involve both agent parameters and environment parameters. The complexity of the model can also be visualized in the hypergraph displays shown in the Supplemental Figures.

3) While there are many strong agent-environment parameter interactions, still the agent parameters taken all together are more predictive (21.0% uncertainty reduction) than the environment parameters taken all together (8.3% uncertainty reduction) (Table 6).

4) Just as in the simplest RA analysis one can rank individual parameters by their predictive efficacy, one can also rank the 16 relations in the best BIC model by their predictive efficacy (Table 7). Not surprisingly, the four most predictive relations all involve the top two predicting parameters, Pop and Vis, which are supplemented by one additional parameter, either an agent parameter or an environmental parameter.

5) RA analysis also tells us what the predictions actually are for all possible states of individual parameters, multi-parameter relations, or multi-relation models. However, there are too many states for the 10 parameters in the BIC model to show predictions for all of them, so Table 9 shows the predictions for states of only the single most predictive relation in the model, namely PopVisMaxgrGini. Predictions are expressed as conditional probability distributions. Table 9 also shows how many times more likely the predicted Gini is than Gini=1, the lowest

income inequality, and Gini=4, the highest income inequality. Figure 4 summarizes the conditional probabilities in an easy-to-grasp decision tree.

OCCAM thus provides a detailed analysis of the relations between the parameters in the NetLogo model and indicates which parameters and interactions among parameters are influential in determining the emergent properties of the simulation. Such detailed exploration of the parameter space is not only interesting as a simulation post-processing. It is also interesting in providing new perspectives to the simulation designer, and it can additionally be useful for determining the extent to which an implemented model corresponds to a conceptual model and has realistic outputs. The potential use of OCCAM as an exploratory tool for NetLogo and other ABS packages also offers a promising mode of exploring other model validation procedures including microvalidation, macrovalidation, and empirical validation as described by Wilensky and Rand (Wilensky and Rand 2015).

Discussion and next steps

This sensitivity analysis of Wilensky's Wealth Distribution Model is intended to be portable to other simulation models, thereby adding another model analysis tool to the modeler's toolkit. Another approach could have used the Behavior Search feature in NetLogo, however the output of that process would require a considerable degree of statistical analysis and interpretation. The approach demonstrated here directly provides the interpretable results for the modeler.

Additionally, OCCAM is well suited for studying different agent-based models for equivalence. Two models are considered to be approximately equivalent if both produce similar distributions of results that cannot be distinguished statistically or the results of the two models produce the same internal relationships (Axtell, et al. 1996). These two categories of equivalence could be tested simultaneously and very quickly using the OCCAM comparison feature which allows the user to perform and compare the same search sequence on two data files.

An additional OCCAM capability that can provide more detailed examination of an agent-based simulation model is the state-based search and state-based fit, in which state-based RA considers many more models using a finer granularity of the Lattice of Structures, where the the number of structures in this lattice is affected by the cardinalities of the variables (Zwick 2019).

Further investigation on the roles of fitness or wealth inheritance and population carrying capacity under resource redistribution (trading and markets) by adding these features to the model and testing the sensitivity of the Gini coefficient to changes in such features could lead to new insights regarding wealth distributions and sustainability in simple economies. Additionally, altruistic rules could be given to some agents in order to study how wealth might be redistributed without the coordinating role of a central authority.

Robert Axelrod stated, “Perhaps the most useful outcome of a simulation model is to provide new ways of thinking about old problems” (1996). Since the inception of computer simulation models like agent-based models, there have been challenges in testing the sensitivity of model outcomes to initial conditions and parameter settings (Epstein and Axtell (1996). Data-mining and machine learning applications offer a new approach for exploring the relations between the model parameters and the model outcomes. Machine learning analyses such as this expands the modelers’ options in the verification and validation process of building and testing agent-based simulation models, and provide insight into how macro system properties, such as wealth inequality, emerge from micro agent-environment interactions.

References

Ashby, W. R. (1964). Constraint Analysis of Many-Dimensional Relations. *General Systems Yearbook*, vol. 9, pp. 99-105

Ashby, W. R., Goldstein, J.(2011). Variety, Constraint, and the Law of Requisite Variety. *Emergence: Complexity and Organization* vol. 13(1-2), 190-207. Originally published as Ashby, W.R. (1968). Variety, constraint, and the law of requisite variety. *Modern Systems Research for the Behavioral Scientist*, (W. Buckley, ed.), Aldine Publishing, Chicago

Axtell, R. L., Axelrod, R. M., Epstein, J. M., Cohen, M. D. (1996) Aligning Simulation Models: A case study and results. *Computational & Mathematical Organization Theory*, 1(2), 123-141

Axelrod, R. M. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press.

Epstein, J. M., Axtell, R. L. (1996) *Growing Artificial Societies: Social Science from the Bottom Up*. The MIT Press. Cambridge.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, 4(5), 41-60.

Jones, B. (1985) Reconstructability Analysis for General Functions. *International Journal of General Systems*, vol 11, 133-142

Klir, G. J. (1985) *Architecture of Systems Problem Solving*. Plenum Press. New York.

Klir, G. J., & Way, E. C. (1985). Reconstructability analysis: aims, results, open problems. *Systems Research*, 2(2), 141-163.

Klir, G. J. (1986) Reconstructability analysis: An offspring of Ashby's Constraint Theory. *Systems Research*, 3(4), 267-271

Klir, G. J. (2006) *Uncertainty and Information: Foundations of Generalized Information Theory*. John wiley & Sons. Hoboken, New Jersey

Knoke, D., Burke, P. J. (1980) *Log-linear Models*. Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 20. Sage Publications. Newbury Park, London, New Delhi

Krippendorff, K. (1986) *Information Theory: Structural Models for Qualitative Data*. Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 62. Sage Publications. Newbury Park, London, New Delhi

Resnick, M. (1994) *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Massachusettes Institute of Technology Press, Cambridge, MA

Shannon, C. E. (1948) A Mathematical Theory of Communication. *The Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, July, October 1948

Simon, H. A. (1991) Bounded rationality and organizational learning. *Organization Science*, 2(1), 125-134.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. (1998). NetLogo Wealth Distribution model. http://ccl.northwestern.edu/netlogo/models/WealthDistribution. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U., Rand, W. (2015) *An Introduction to Agent-Based Modeling* The MIT Press. Cambridge.

Willett, K., and Zwick, M. (2003). A Software Architecture for Reconstructability Analysis. *Kybernetes*, 33(5/6), 877-905.

Zwick, M. (2001)." Wholes and Parts in General Systems Methodology," The Character Concept in Evolutionary Biology. (Gunter Wagner, ed). *Academic Press*. DOI: 10.1016/B978-012730055-9/50020-3

Zwick, M. (2004). An overview of reconstructability analysis. *Kybernetes*, 33(5/6), 877-905.

Zwick, M. (2019). OCCAM: A Reconstructability Analysis Program Manual version 3.4.0 available at https://www.pdx.edu/sysc/sites/www.pdx.edu.sysc/files/woccaman-3.1.2019_mz.pdf .Information about reconstructability analysis and access to OCCAM online is available at https://www.pdx.edu/sysc/research-discrete-multivariate-modeling . Access to open-source version is available at https://www.occam-ra.io/.

Zwick, M. (2020). Reconstructability Analysis and its OCCAM implementation. International Conference on Complex Systems (New England Complex Systems Institute) 2020, July 29. https://works.bepress.com/martin_zwick/166/download/, slide 26.

Table 1: NetLogo Wealth Distribution model variables with descriptions

Variable Name	Definition and function
Agent parameters	
num-people	Total size of the population. This number does not represent a maximum population size, but rather it is a fixed number of individuals for each simulation. The population size is thus static; it only changes when the parameter value (using the NetLogo interface slider) is changed.
metabolism-max	Each individual at birth is assigned a number that determines how much grain at each time step is required to stay alive. The maximum number is set by the slider, but each agent is assigned a random number in the range [1, max] for a metabolism value.
max-vision	Similar to metabolism-max, this is a set-point for the maximum vision level, where each individual at birth is assigned a random number in the range [1, max] that determines how far around itself it can see in order to find grain and execute the decision rule to move.
life-expectancy-min	New individual agents in the population will live at least this many time steps.
life-expectancy-max	Individuals live at most this many time steps.
Environment parameters	
grain-growth-interval	How long it takes for grain to grow back once a patch’s resources have been depleted. Low values are associated with more abundant resources because it takes fewer time steps for the patch to recover.
num-grain-grown	How much grain is grown at each time step.
percent-best-land	The initial setting for the density of patches that are seeded with the maximum amount of grain at time = 0.
max-grain	Global variable in the model code tab, default value set to 50. This variable determines the maximum amount of grain any patch can hold at any time during the simulation.

Table 2: Definitions for OCCAM output search measures (Zwick 2019)

OCCAM measure	abbreviation	description
search ID number	ID	unique model identifier assigned during search
model specification	MODEL	specified model where "IV" is a relation with all the independent variables in it
Δ-Degrees of Freedom	ΔDF	difference in degrees of freedom between the model and the reference (the independent model); for the reference, this delta is 0
uncertainty reduction	%ΔH(DV)	percent reduction in uncertainty of the dependent variable for the model
ΔBayesian Information Criterion	ΔBIC	difference of the values for Bayesian Information Criterion between the reference and the model
%correct training data	%C(train)	performance of the model in predicting the training data
percent coverage	%cover	portion of the state space of predictors in the model that is present in the data
%correct test data	%C(test)	performance of the model in predicting the test data

Table 3: NetLogo BehaviorSpace Experimental Design Settings

NetLogo Name	OCCAM abbreviation	low	high
num-people	Pop	100	1000
metabolism-max	Metab	1	25
max-vision	Vis	1	15
life-expectancy-min	MinAge	1	15
life-expectancy-max	MaxAge	50	100
grain-growth-interval	GrInt	1	10
num-grain-grown	GrRate	1	10
percent-best-land	PBLand	5	25
max-grain	Maxgr	50	500
step	Time	26	determined in BehaviorSpace

Table 4: Summary of each variable as a single predictor of the Gini coefficient outcome for the NetLogo wealth distribution model. (The model's predictive relation is listed; its 'IV' relation is omitted.)

ID	Model	ΔDF	Alpha	%ΔH(DV)	ΔBIC	%C(train)	%C(test)
11*	PopGini	3	0.000	6.376	1658.4	46.1	40.5
10*	VisGini	3	0.000	6.280	1632.9	44.1	38.4
9*	GrrateGini	3	0.000	2.939	749.3	44.1	38.4
8*	GrintGini	3	0.000	2.023	507.2	44.1	38.4
7*	MaxageGini	3	0.000	1.930	482.5	44.1	38.4
6*	MetabGini	3	0.000	1.897	473.8	44.1	38.4
5*	MaxgrGini	3	0.000	1.771	440.5	44.1	38.4
4*	PblandGini	3	0.000	0.516	108.6	44.1	38.4
3*	TimeGini	6	0.000	0.454	64.5	44.1	38.4
2	MinageGini	3	0.926	0.002	-27.3	44.1	38.4
1*	Gini	0	1.000	0.000	0.0	44.1	38.4

Table 5: Best Model Summaries from OCCAM Search Results (IV relations are deleted)

Loopless Models Search

	ID	Model	ΔDF	% ΔH (DV)	ΔBIC	%C train	%C test
ΔBIC	78*	PopMetabVisMaxageGrintGrrateMaxgrGini	381	46.2	8688	66.0	55.5
ΔAIC	87*	PopMetabVisMaxageGrintGrratePblandMaxgrGini	765	52.9	6912	70.1	60.0
IncrP	89*	PopMetabVisMaxageGrintGrratePblandMaxgrTimeGini	2301	57.3	-6103	72.5	59.1
%C(test)	87*	PopMetabVisMaxageGrintGrratePblandMaxgrGini	765	52.9	6912	70.1	60.0

Disjoint Models Search

	ID	Model	ΔDF	% ΔH (DV)	ΔBIC	%C train	%C test
ΔBIC	86*	PopGrintGini:MetabGini:VisMaxgrGini:MaxageGini:GrrateGini	27	30.5	7819	59.3	52.7
ΔAIC	84*	PopGrrateGini:MetabVisGini:MaxageGini:GrintGini:MaxgrGini	27	30.5	7804	60.6	54.7
IncrP	82*	PopGini:MetabVisGini:MaxageGini:GrintGini:GrrateGini:PblandGini:MaxgrGini	27	30.4	7789	61.3	54.2
%C(test)	79*	PopGrintGrrateGini:VisGini:MaxageGini:MaxgrGini	30	28.9	7369	59.0	52.4

All Models Search

	ID	Model	ΔDF	% ΔH (DV)	ΔBIC	%C train	%C test
ΔBIC	531	PopMetabVisGini:PopMetabGrrateGini:PopMetabPblandMaxgrGini:PopVisMaxageGini:PopVisGrintGrrateGini:PopVisPblandGini:PopVisMaxgrGini:PopGrratePblandGini:PopGrrateMaxgrGini:MetabVisGrintGini:MetabVisGrrateGini:MetabVisPblandGini:MetabGrintGrrateGini:MaxageMaxgrGini:GrintPblandGini:TimeGini	162	48.2	11248	68.2	60.3
ΔAIC	701	PopMetabVisGini:PopMetabGrratePblandGini:PopMetabPblandMaxgrGini:PopVisMaxageGini:PopVisGrintGrrateGini:PopVisPblandMaxgrGini:PopGrratePblandMaxgrGini:MetabVisGrintPblandGini:MetabVisGrrateGini:MetabVisPblandMaxgrGini:MetabGrintGrrateGini:VisMaxageMaxgrGini:VisGrratePblandMaxgrGini:MaxagePblandMaxgrGini:TimeGini	213	49.2	11046	68.5	60.3
IncrP	530*	PopMetabVisGini:PopMetabGrrateGini:PopMetabPblandMaxgrGini:PopVisMaxageGini:PopVisGrintGrrateGini:PopVisPblandMaxgrGini:PopGrrateMaxgrGini:MetabVisGrintGini:MetabVisGrrateGini:MetabVisPblandGini:MetabGrintGrrateGini:MaxageMaxgrGini:GrintPblandGini:TimeGini	162	48.1	11226	67.9	60.1
%C (test)	551	PopMetabVisGini:PopMetabGrrateGini:PopMetabPblandMaxgrGini:PopVisMaxageGini:PopVisGrintGrrateGini:PopVisPblandMaxgrGini:PopGrratePblandGini:PopGrrateMaxgrGini:MetabVisGrintGini:MetabVisGrrateGini:MetabVisPblandGini:MetabGrintGrrateGini:MaxageMaxgrGini:GrintPblandGini:TimeGini	168	48.4	11239	68.4	60.7

Table 6: Partitioning Uncertainty Reduction by Variable Type

description	IV count	ΔDF	%ΔH(DV)	ΔBIC	%C(train)	%C(test)
All variables	10	162	48.2		68.2	60.3
Agent IVs (Pop, Metab, Vis, Minage, Maxage)	5	33	21.0	5258	54.5	47.9
Agent IVs omitting Pop (Metab, Vis, Minage, Maxage)	4	15	12.3		46.5	40.6
Environment IVs (Grit, Grrate, Pbland, Maxgr)	4	27	8.3	1890	46.7	38.7
Other IV (Time)	1	6	0.5	65	47.5	45.6

Table 7: Percent Correct by Model for each Relation and model improvement in best all-models ΔBIC model

Relation	%C(test)	%Improvement
PopMetabVisMinageMaxageGrintGrratePblandMaxgrTime	60	56
PopVisMaxgrGini	50	82
PopMetabVisGini *(all agent predictors)*	49	94
PopVisMaxageGini *(all agent predictors)*	49	100
PopVisPblandGini	49	97
PopGrrateMaxgrGini	47	97
PopVisGrintGrrateGini	46	61
PopMetabPblandMaxgrGini	46	62
MetabVisGrrateGini	42	63
MetabVisPblandGini	41	56
MetabVisGrintGini	41	45
PopMetabGrrateGini	41	41
PopGrratePblandGini	40	26
MetabGrintGrrateGini	38	3
MaxageMaxgrGini	38	0
GrintPblandGini *(all environmental predictors)*	38	0
TimeGini	38	0

Table 8: Table of variables present in relations of best all-models ΔBIC model

Relation	Pop	Vis	Grrate	Grint	Maxage	Metab	Maxgr	Pbland	Time	Number of variables in relation
PopMetabVisGini	1	1				1				3
PopMetabGrrateGini	1		1			1				3
PopMetabPblandMaxgrGini	1					1	1	1		4
PopVisMaxageGini	1	1			1					3
PopVisGrintGrrateGini	1	1	1	1						4
PopVisPblandGini	1	1						1		3
PopVisMaxgrGini	1	1					1			3
PopGrratePblandGini	1		1					1		3
PopGrrateMaxgrGini	1		1				1			3
MetabVisGrintGini		1		1		1				3
MetabVisGrrateGini		1	1			1				3
MetabVisPblandGini		1				1		1		3
MetabGrintGrrateGini			1	1		1				3
MaxageMaxgrGini					1		1			2
GrintPblandGini				1				1		2
TimeGini									1	1
Variable Frequency	9	8	6	4	2	7	4	5	1	

Table 9: OCCAM Fit Output example of most predictive relation within the best all-models ΔBIC model: conditional probabilities for DV states given IV states

IV state			p(DV|IV) as percentage				prediction & risk ratios		
Pop	Vis	Maxgr	Gini=1	Gini=2	Gini=3	Gini=4	rule	Ratio_G1	Ratio_G4
100	1	50	15.08	23.75	36.8	24.38	3	0.91	1.54
100	1	500	3.59	8.52	20.78	67.11	4	0.22	**4.24**
100	15	50	33.83	34.14	24.3	7.73	2(1)	**2.04**	0.49
100	15	500	26.88	35.16	35.7	2.27	3(2)	1.62	0.14
1000	1	50	8.67	81.8	8.13	1.41	2	0.52	**0.09**
1000	1	500	3.75	47.58	28.75	19.92	2	0.23	1.26
1000	15	50	24.22	57.11	15.08	3.59	2	1.46	0.23
1000	15	500	16.48	64.61	18.59	0.31	2	1	**0.02**
		Marginal values	16.56	44.08	23.52	15.84	2		

List of Figures

Figure 1: OCCAM Loopless Search results with ΔDF plotted on log10 scale on X-axis and Percent Uncertainty Reduction %ΔH(DV) on Y-axis

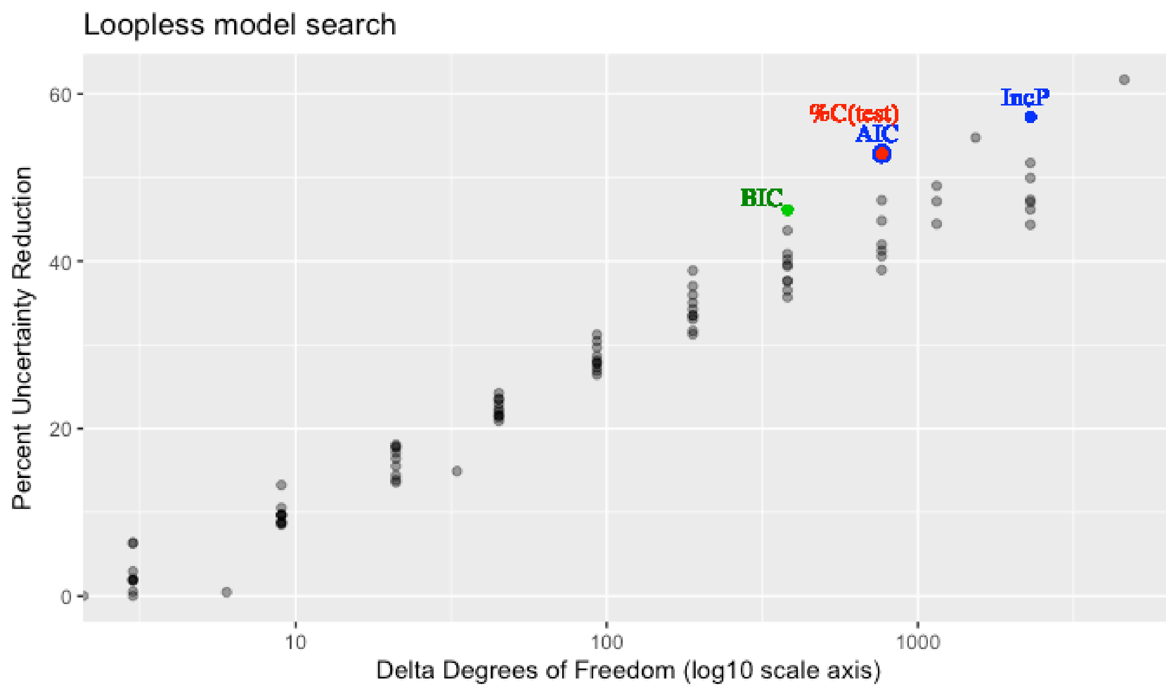


Figure 2: OCCAM Disjoint Search results with ΔDF plotted on log10 scale on X-axis and Percent Uncertainty Reduction %ΔH(DV) on Y-axis

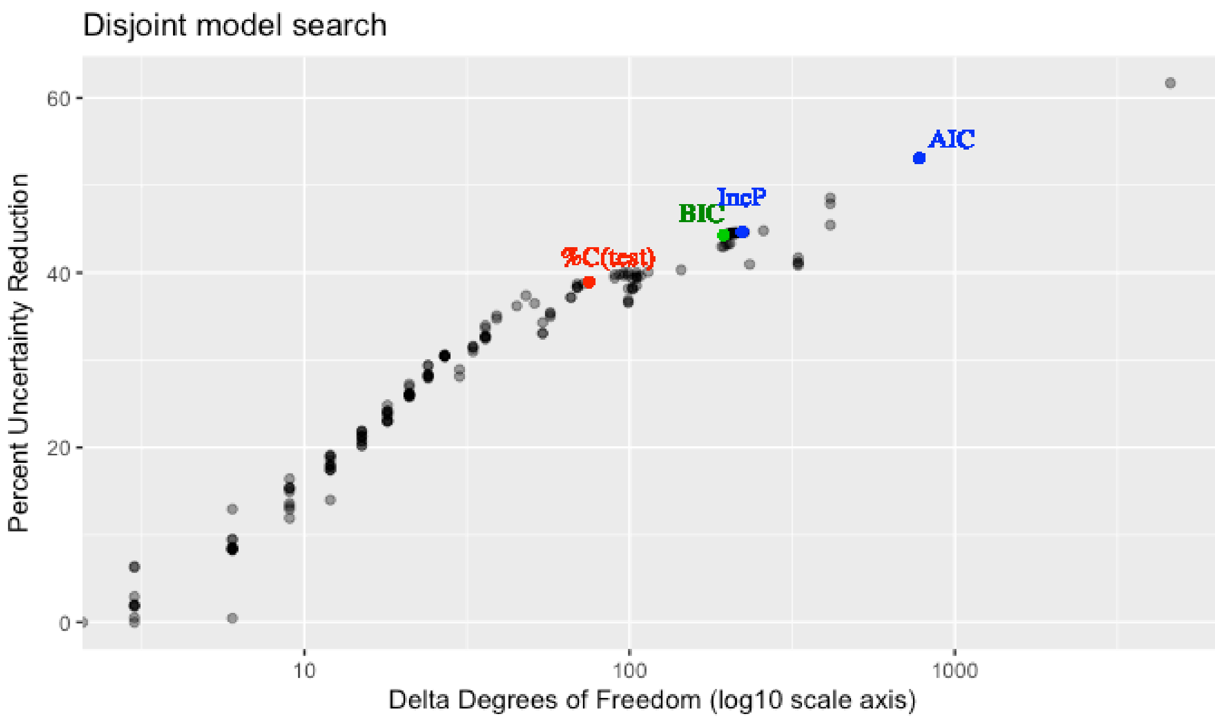


Figure 3: OCCAM All-Models Search results with ΔDF plotted on log10 scale on X-axis and Percent Uncertainty Reduction %ΔH(DV) on Y-axis

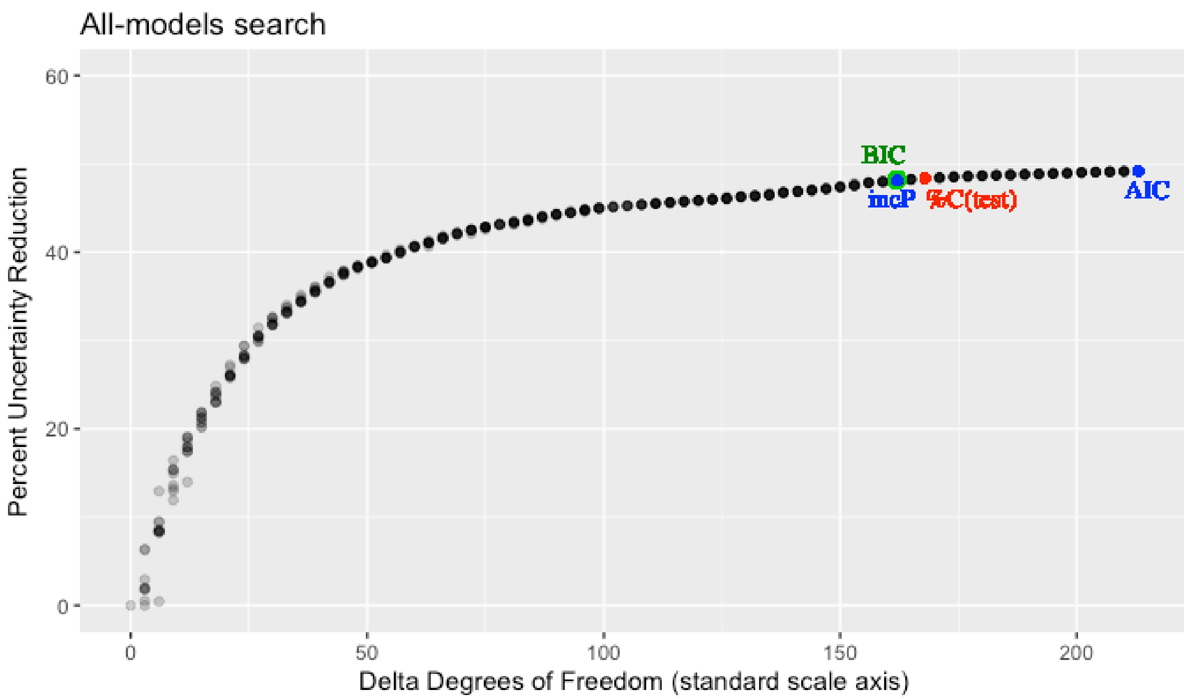
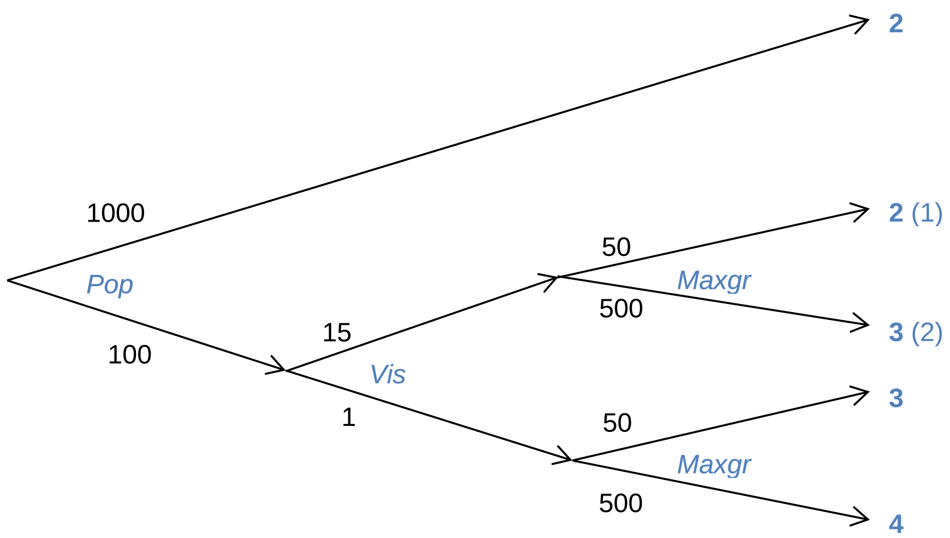


Figure 4: Decision Tree indicating predicted DV state for the most predictive Fit relation (*PopVisMaxGr*) from the best model by ΔBIC results of the OCCAM all-models search. For (Pop, Vis) = (100, 15), a second predicted state is shown in parentheses; this state is only slightly less probable than the state not in parentheses.



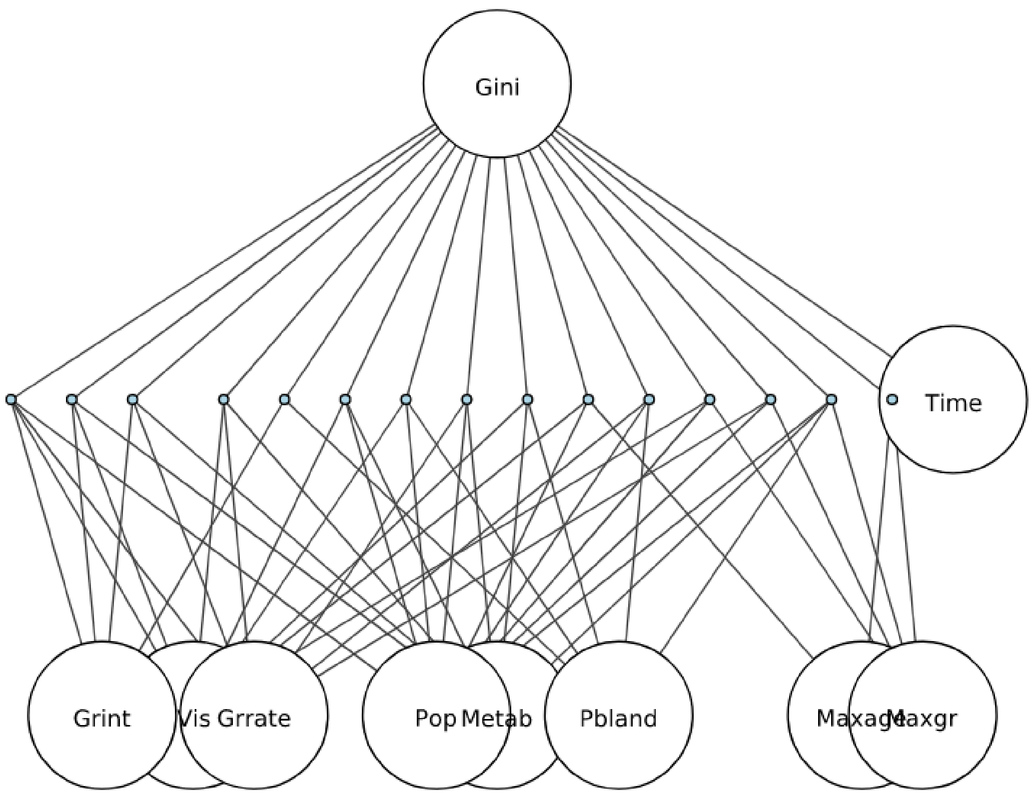
Supplemental Figures

Hypergraphs of all-models search result best *Δ*BIC model

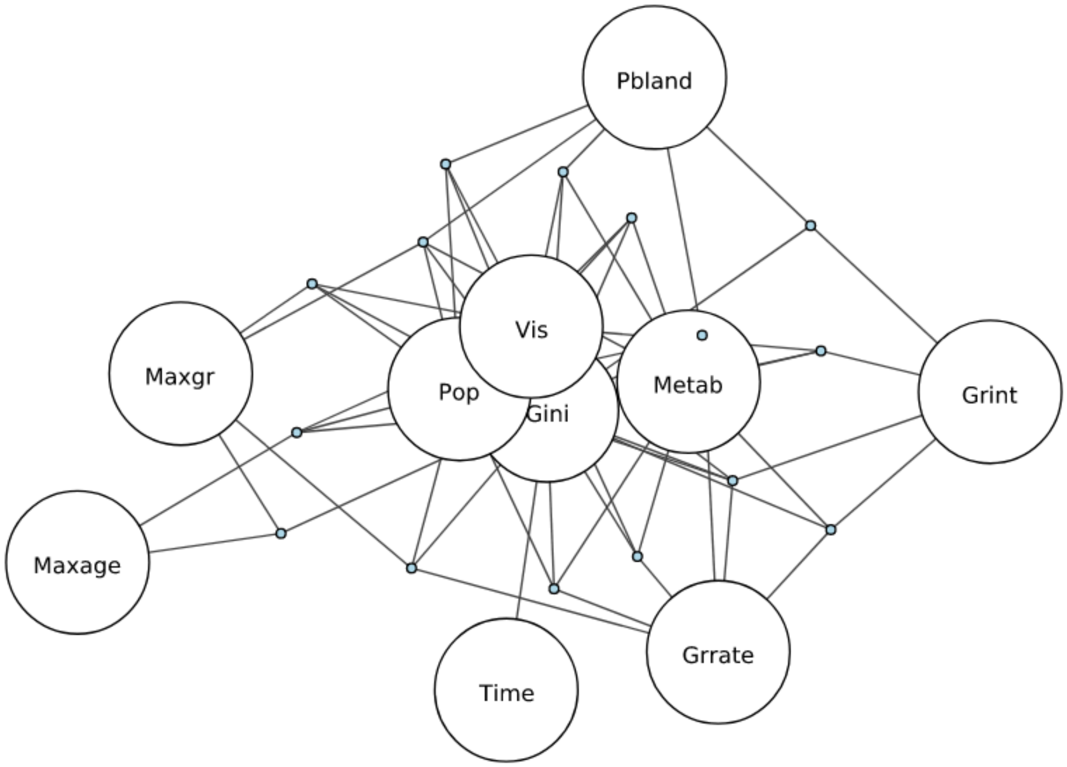
ΔBIC Model

IV:PopMetabVisGini:PopMetabGrrateGini:PopMetabPblandMaxgrGini:PopVisMaxageGini:PopVisGrintGrrateGini:PopVisPblandGini:PopVisMaxgrGini:PopGrratePblandGini:PopGrrateMaxgrGini:MetabVisGrintGini:MetabVisGrrateGini:MetabVisPblandGini:MetabGrintGrrateGini:MaxageMaxgrGini:GrintPblandGini:TimeGini

Sugiyama layout



Kamada-Kawai layout



Fruchterman-Reingold layout