Agent-Based Modeling and Simulation

Stochasticity

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

`mailto:aguerra@uv.mx`
`https://www.uv.mx/personal/aguerra/abms`

Doctorado en Inteligencia Artificial 2024

Universidad Veracruzana

Credits

- These slides are based on the book of Railsback and Grimm [4], chapter 6.
- Any difference with this source is my responsibility.
-
-

Universidad Veracruzana

Ideas

- In modeling, the word stochastic describes processes that are at least partly based on random numbers or events.
- The word deterministic is used as the opposite of stochastic, describing processes that are modeled with no randomness so they produce exactly the same result each time they are executed.
- Many ABMs have relatively few stochastic processes and their most interesting and complex results arise from unique individuals executing deterministic behaviors in a variable environment.

Standard techniques

- When we represent some source of variability in our model as stochastic, as a sequence of random numbers, we do three things:
    - Select an appropriate statistical distribution for the random numbers;
    - Select the distribution's parameters, or model how the parameters vary during the simulation; and
    - After the model is built, use replication to understand how much of the variability in its results is due to its stochastic processes.
- Traditional modeling literature offers an extensive coverage of stochasticity.

Universidad Veracruzana

Objectives I

- Understand why and how modelers use random numbers, and how we determine the effect of stochasticity on our model results.
- Know what a random number distribution is, and the difference between continuous and discrete distributions.
- Know the properties of uniform, normal, Bernoulli, and Poisson distributions and how to draw random numbers from these distributions.
- Understand use of random generator seeds to control the random numbers produced by NetLogo.

Universidad Veracruzana

Objectives II

- Learn the very common technique of modeling agent behaviors as stochastic events with random number distributions parameterized from data.
- Practice defensive programming to catch and correct run-time errors.
- Become aware that there are many recipes in the literature to use when you must write your own code for numerical methods such as random number distributions.

Universidad Veracruzana

Ideas

- Why are random numbers and stochastic processes so widely used and important in simulation modeling?
- To represent some kind of variability because it exists in reality and may have important effects, but avoiding to explain why that variability occurs.
- One way to simplify variables or processes that change over time or space is to assume that they are random.
- Then, instead of having to model what causes those variables or processes to change, we simply draw random numbers to make them change.

Universidad Veracruzana

Initializing Variables

- One of the most common uses of stochasticity is to assign initial values to variables, especially when initializing a model.
- Example. An economic model might treat households as agents having a variable for net worth. When the model starts, each agent is given a unique but realistic net worth. Suppose data indicate the net worth values of real households follow a normal distribution with a mean of $223K and a standard deviation of $83K. Then each agent is given a net worth drawn randomly from a normal distribution with this mean and standard deviation.

Universidad Veracruzana

Variable outcomes I

- Another use of stochasticity is to model processes that produce variable outcomes when we do not want to represent those processes in great detail.
- To produces realistic results, we can parameterize a stochastic process with data on the rate or frequency of real events.
- Example. Imagine a model (of crop or livestock production, or forest fires, or tourism) that depends on when it rains. Instead of modeling the weather, which is extremely complex, we can simply examine some weather data, *e.g.*, treat the observed frequency of rainfall as a probability of it raining on any day in our model.

Universidad Veracruzana

Variable outcomes II

- On each simulated day, we can get a random number between zero and one; if this number is less than the probability of rainfall, then our model experiences rain.

Universidad Veracruzana

Variable Behavior

- Treating observed frequencies as probabilities in a stochastic process can also be used to model agent behavior.
- Example. An anthropological model might simulate how often a prehistoric family moves to a new location. If there are archeological data indicating how frequently real families moved, then the observed frequencies could be used as the probability of a model family moving on any particular time step.

Universidad Veracruzana

Conditional Probabilites

- Previous examples of stochastic process models are very simple and leave out contingencies, *i.e.*, how probabilities depend on other variables, that could be important.
- Example. Even prehistoric people must have been smart enough to move not just at random times but more often when they had a good reason.
- It is best to start our models as simply as possible, but sometimes we clearly need to add some of these contingencies.
- Statistical techniques, *e.g.*, logistic regression, model how the probability of some event varies with one or several variables.
- Bayesian Networks [5, 1] are a more interesting alternative for this.

Universidad Veracruzana

Observations

- Still, we must realize that by representing a behavior via a probability distribution we are at least partly imposing its outcomes.
- Therefore, behavior models often combine contingencies that represent adaptive decisions, *e.g.*, in the form of if-then rules, with stochasticity.

Question

- When using stochasticity to represent variables or processes and behaviors in a model, we raise the question of how random our results are.
- Example. If an ABM is very simple and has only one major environment variable that is modeled stochastically, will the results just be entirely random?
- Example. Or if our ABM uses many stochastic processes, will it always produce nearly the same results because the effects of its parameters and inputs are swamped by all the randomness?

Replication I

- These questions are the reason we use replication, *i.e.*, running the model multiple times, with the random numbers being the only thing that changes.
- The results from a replication experiment only describe the effects of stochasticity in our model, and nothing else.
- Replication does not tell us how much uncertainty there is in a model, because uncertainty comes not just from stochastic processes but also from uncertainty in parameter values, inputs, and assumptions.
- Variability from a replication experiment should not be compared to the real one, unless we assume that the stochastic processes in the model represent all sources of variability in the preal system.

Universidad Veracruzana

Replication II

- Remember that statistical tests that depend on the number of replicates, *e.g.*, t-tests, must be used carefully because the number of replicates is arbitrary.
- Example. A change too small to be scientifically meaningful (compared to all the sources of uncertainty in a model and in data on the real system) can be statistically significant if we run enough replicates.

Universidad Veracruzana

Replication uses

- Replication is appropriate, and essential, for comparing different versions of a model to each other.
- Example. When we change parameter values, equations, assumptions, or input data and want to know what the effect is.
- The best way to analyze such replication experiments is often to simply choose a reasonable number of replicates, *e.g.*, 10 for a simple model, and graph the range of results.

Universidad Veracruzana

Pseudo Random Numbers in NetLogo

- We have some experience with stochastic processes, having used the primitives `random` and `random-float` to draw pseudo random numbers from a certain range.
- Pseudo refers to the fact that these numbers are not truly random, *i.e.*, they are generated by a mathematical algorithm (the Mersenne Twister in NetLogo 6.4) deterministically.
- Since they are commonly used in computer science, we use both terms indistinguishably.

Universidad Veracruzana

Random Number Generator I

- The Random Number Generator of NetLogo has been rigorously tested to produce results that very closely resemble a truly random process.
- However, it is possible to reproduce exactly the same sequence of pseudo random numbers by setting the generator's seed, *i.e.*, an integer in the range -2147483648 to 2147483647.
- Example. If you run these commands:

```
random-seed 137
show random 100
show random 100
show random 100
```

You will always get 79, 89, and 61 in that order.

Random Number Generator II

- This let us eliminate the random numbers as a source of variability among model runs.
- You can use the reporter `new-seed` to create a seed evenly distributed over the space of possible seeds, based on the current date and time.
- If you don't set the random seed yourself, NetLogo sets it to a value based on the current date and time. The random seed is unknown in this case.
- Many operations make use of these numbers for doing random choices, *e.g.*, `one-of`, `sprout`, `uphill`, etc.
- Review the Programming Guide to more details on Random Numbers.

Universidad Veracruzana

Ideas

- NetLogo provides different random distributions, selecting the appropriate one can be a key part of designing a model.
- If you use random numbers for any but the simplest purposes, you should make yourself familiar with the basic probability theory behind common distributions so you understand their assumptions and use them appropriately.
- Modelers typically use two distinct types of distributions:

Continuous.	Produce real numbers in the mathematical sense: they include fractions and can be any value over some range.
Discrete.	Produce only certain discrete values such as true or false, or integers.

Universidad Veracruzana

Continous Uniform Distribution

- A uniform real number distribution (`random-float`) simply provides random floating point numbers between a specified minimum and maximum, with all values within that range equally likely.
- When modelers talk in general about a random number they refer to a uniform distribution with minimum 0.0 and maximum 1.0.
- Such random numbers are useful for modeling whether an event occurs from a parameter that represents the probability of the event occurring.
- Example. The probability of a turtle dying on each tick can be used as follows:

```
if random-float 1.0 < mortality-prob [die]
```

Continuous Normal Distribution

- A normal distribution (`random-normal`) produces a sequence of random floating point numbers that create the familiar bell curve, with approximately 68% of the numbers falling within a range of plus-or-minus 1 standard deviation of the mean, and approximately 95% of numbers within 2 standard deviations of the mean.
- To use this distribution, we must specify the mean and standard deviation.
- Example. To give each patch a value for some variable called `food-level`:

```
ask patches [set food-level random-normal 5.0 1.5]
```

Universidad Veracruzana

Observations I

- While a normal distribution produces values more likely to be closer to its mean, it is important to understand two of its other characteristics:
  1. Unlike a uniform distribution, there is no limit on the range of values a normal distribution can produce, so
  2. When you draw many values from a normal distribution, it is quite likely that a few will be so far from the mean that they are nonsensical for the model.
- Example. In the above example, a negative value of foodlevel probably makes no sense and could cause mathematical errors or absurd behavior in the model.

Universidad Veracruzana

Observations II

- NetLogo version 6.4 implements also `random-exponential`, `random-gamma`, `random-poisson`.

Defensive Programming I

- What can you do about the possibility that random normal values are occasionally nonsensical?
- Write your NetLogo program so it checks for nonsensical values from random-normal and does something about them.
- Example. Specify a minimum value, such as 0.5:

```
ask patches [set food-level random-normal 5.0 1.5
    if food-level < 0.5 [set food-level 0.5]
]
```

Universidad Veracruzana

Defensive Programming II

- Or you can have your program simply draw another random number if it obtains a negative one. A while loop is useful for this:

```
ask patches [
  let my-rand -1.0
  while [my-rand <= 0.0] [
    set my-rand random-normal 5 1.5
    ; show my-rand ; Test output
  ]
  set food-level my-rand
]
```

- If you use these tricks you are changing the distribution of values actually used in your model; the mean of the values you keep will be different from the mean you specify in the `random-normal` statement.

Universidad Veracruzana

Other Continuous Distributions

- Another way to avoid negative values is to use a log-normal distribution instead of a normal distribution.
- The exponential distribution is often useful because it creates skewed results: many small values and few high values, with the probability of a value being drawn decreasing as that value increases.
- Example. The number of people with different levels of wealth.
- The exponential distribution never produces zero or negative values.
- The gamma distribution is commonly used in modeling but because it is more complex we do not discuss it.

Universidad Veracruzana

Ideas

- Discrete random number distributions are important because ABMs are inherently discrete integer models.
- Many of the processes we model use true-false choices (Does a turtle execute some behavior or not? Does it die or not?) and integer variables: the number of agents that do something (die, move, etc.), the number of patches a turtle can sense, the number of new turtles created by reproduction, and so on.
- Example. Above we used a uniform real distribution to model when turtles die:

```
if random-float 1.0 < mortality-prob [die]
```

Universidad Veracruzana

Bernoulli Distribution

- The Bernoulli distribution returns a boolean for one trial of an event, when the probability of that event being true is provided as a parameter.
- Example. The previous code can be re-implemented as follows:

```
if random-bernoulli mortality-prob [die]
```

- The problems is that `random-bernoulli` is not provided by NetLogo.

Universidad Veracruzana

Implementation

- You can implement the following reporter:

```
to—report random-bernoulli [probability-true]
  ; First, do some defensive programming to make
  ; sure "probability-true" has a sensible value

  if (probability-true < 0.0 or probability-true > 1.0) [
    user-message (word
                   "Warning in random-bernoulli: probability-true equals"
                   probability-true)
  ]
  report random-float 1.n0 < probability-true
end
```

Discrete Uniform Distribution

- The primitive `random` is a uniform integer distribution: it produces integers that are evenly distributed over a specified range.
- This primitive assumes that one end of the range is always 0.
- Example. If we want a turtle to create a number of offspring that is uniformly random between 2 and 6, we would tell the turtle to:

```
hatch 2 + random 5
```

Universidad Veracruzana

Poisson Distribution I

- It models the number of times that some event occurs over a specified interval (time or space), when the average rate of occurrence is known and when the events are independent of each other.
- The primitive `random-poisson` takes a parameter that is equal to:
  1. The average rate (mean number of events per unit of time or space), multiplied by
  2. The interval length (number of units of time or space being modeled).
- Example. If we know that patches sprout new turtles at an average rate of 0.4 times per tick, then we can tell the patches to:

```
sprout random-poisson 0.4
```

Universidad Veracruzana

Poisson Distribution II

- Example. If instead we want to model how many turtles are sprouted by all the patches, we can still use `random-poisson`:

```
let number-sprouts random-poisson (0.4 * count patches)
```

Universidad Veracruzana

Binomial Distribution

- It models the number of events that occur when there are n chances for an event to occur and the probability of each event occurring on a given chance is P.
- Another way to think of the distribution is this: if you execute the Bernoulli distribution n times, how many true values do you get?
- There is no primitive for this distribution in NetLogo.
- But you can use extensions

Extensions

- Dist. Makes it easy to plot frequency distributions and complementary cumulative distribution functions in NetLogo. Don't use in production code.
- R. Provides primitives to create R-Variables with values from NetLogo variables or agents and others to evaluate commands in R with and without return values
- rngs. Allows for the definition of multiple independent streams of random numbers as well as an expansion of available discrete and continuous distributions. Based on the colt library.

Universidad Veracruzana

Local Randomness I

- BehaviorSpace does not override any `random-seed` statements in your program, *e.g.*, at the beginning of the `setup` procedure.
- Example. BehaviorSpace will produce 10 identical replicates.
- If you want to produce replicate simulations, remove any such `random-seed` statements.
- What if we want one part of a model to be random, but consistent among model runs (including BehaviorSpace replicates), while the rest of the model varies stochastically each time the model runs?

Universidad Veracruzana

Local Randomness II

- Example. We might want a patch variable `resource-level` initialized to random values from a normal distribution with a mean of 100 and a standard deviation of 20, but we want these initial values, and only them, to be exactly the same each time the model runs:

```
to setup
  with-local-randomness [
    random-seed 131313
    ask patches [set resource-level random-normal 100 20]
  ]
  ...
end
```

The Businnes Investor Model

- Agents in this model (chapter 10.4) make one important adaptive decision: they decide each time step whether to switch to one of the other investment opportunities (patches) available to them.
- These investment alternatives may provide higher or lower profit and may have higher or lower risk.
- The agents make this decision by selecting the alternative that maximizes an objective function that represents expected investor wealth at the end of a time horizon.

Universidad Veracruzana

Data

- Now, imagine that instead of relying on this theoretical approach to modeling investor behavior, we had data from real investors making similar decisions.
- How could we use these data in an empirical submodel, one that imposes observed decision patterns on the model?
- Let's assume we have the observations from 12,500 separate decisions by real investors –a seemingly large data set–
- These observations are actually results from 10 runs of the investor model from section 10.4.

Universidad Veracruzana

Observed Frequencies of Investor Decisions

Alternative	Times Available	Times Chosen	Freq
Higher profit and lower risk	202	162	80%
Higher profit and higher risk	4830	224	4.6%
Lower profit and lower risk	8340	263	3.2%
Lower profit and higher risk	12,174	0	0%
Do not change investment	12,500	11,831	95%

Universidad Veracruzana

Assumptions

- we can assume that the observed frequencies are equal to probabilities that agents will choose the four alternatives when available.
- Example. We assume that investors will change to an investment with higher profit and lower risk with a probability of 0.80 when at least one such investment is available, to an investment with high profit and higher risk with probability 0.046, and so on.
- Using if-then logic and stochastic decisions like this is sometimes called rule-based modeling.
- By the way, there is a CLIPS v6.3 extension for NetLogo.

Universidad Veracruzana

Limitations

- While 12,500 observations seems like a big data set, a number of questions remain unrepresented by the data.
- Example. What did the investors do when more than one alternative was available to them –if they could move to either higher profit and higher risk or lower profit and lower risk, how often did they choose one over the other?
- Example. We also have learned nothing about how the magnitudes of investment risks and returns affect the probability of switching: we might strongly suspect that investors would be more likely to switch to an investment with higher risk if the increase in profit was big instead of small

Universidad Veracruzana

Implementation

```
; First identify potential neighbor destination patches
let potential-destinations neighbors with
  [not any? turtles-here]

; Identify any alternative patches with higher profit and
; lower risk
set HiProfit-LowRisk-alts potential-destinations with
  [(annual-profit > [annual-profit] of myself) and
   (annual-risk < [annual-risk] of myself)]

; Decide whether to move to one, using global
; probability parameter
if any? HiProfit-LowRisk-alts [
  if random-bernoulli P-move-to-higher-profit-lower-risk [
    ; Now move there
    move-to one-of HiProfit-LowRisk-alts
    stop ; Turtles can only move once per tick!
  ]
]

; Repeat for higher profit and higher risk, etc.
```

Universidad Veracruzana

Stochastic Analysis of the Model

- You will need four such blocks of code for the different situations: whether there are any alternatives available with higher profit and higher risk, with lower profit and lower risk, etc.
- Exercise. Now, you might think that this model should perfectly reproduce observed behavior of real investors. But does it? We leave analysis of this stochastic model as an exercise.

Universidad Veracruzana

Summary I

- ABMs are not necessarily stochastic, very interesting and complex results can emerge from models that are completely deterministic.
- Yet, very often, we want some form of variability because we believe it has important effects, but do not want to have to explain its source.
- Stochastic processes are how we represent such variability.

Universidad Veracruzana

Summary II

- When we design a stochastic element, we need to select an appropriate distribution:
  - Do we need the random numbers to be floating point numbers, integers, or boolean values?
  - Do we want the numbers to be uniformly distributed over a discrete range?
  - Do we want them to be normally distributed with some values more likely than others and an extremely broad range of possible values?
  - D we need the values to be positive?

Universidad Veracruzana

Summary III

- Each distribution has its own kinds of parameters: a uniform distribution requires the minimum and maximum of the range of values it can produce; a normal distribution requires a mean and standard deviation, and so on.
- These parameters are often treated as constants that are estimated from data: the frequency with which events have been observed in real systems; or the mean and standard deviation of field measurements of some variable are used as the parameters of a normal distribution in the model.
- It is also very common and useful for the parameters of a statistical distribution to be modeled, *e.g.*, the parameters can vary as a function of whatever is going on in the ABM.

Universidad Veracruzana

Summary IV

- Sometimes we want to turn off some or all of the randomness in a model, by keeping the random numbers from changing every time we run it.
- We can do this by setting the random number seed.
- Via the `with-local-randomness` primitive, NetLogo even lets us switch randomness off and on in different parts of a model.

Universidad Veracruzana

Conclusions I

- The continuous distributions that many people are most familiar with, especially the normal distribution, are not the most useful in ABMs because the processes we model are often best represented by integers or boolean values, not real numbers;
- Or because the range of usable values is limited, *e.g.*, negative sizes or ages often make no sense, but normal distributions are likely to produce negative values.
- Sometimes a distribution can be useful even when its theoretical assumptions are not completely met.
- Example. The Poisson distribution cann often be useful for providing random integers even when its assumption of independence among events is not realistic.

Universidad Veracruzana

Conclusions II

- Whenever you aren't completely sure what numbers a distribution will produce, write a test code and investigate.
- Defensive programming, *i.e.*, putting checks in the code to see if variables have impossible or unrealistic values, is critical for keeping models from producing nonsense without our realizing it.
- It is particularly appropriate when working with stochastic processes, and especially when using distributions like the normal distribution that can occasionally produce outlier values.
- There are potentially useful random number distributions that are not built into NetLogo. Check the extensions before programming them by yourself.

Universidad Veracruzana

Conclusions III

- Forsyth [2] provides a nice introduction to probability and statistics for Computer Science.
- Horgan [3] does the same using R.

Referencias I

[1] AB Downey. *Think Bayes: Bayesian Statistics in Python*. Second. Sebastopol, CA, USA: O'Reilly Media, Inc., 2021.

[2] D Forsyth. *Probability and Statistics for Computer Science*. Cham, Switzerland: Springer, 2018.

[3] JM Horgan. *Probability with R: An Introduction with Computer Science Applications*. Second. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2020.

[4] SF Railsback and V Grimm. *Agent-Based and Individual-Based Modeling*. Second. Princeton, NJ, USA: Princeton University Press, 2019.

[5] M Scutari and JB Denis. *Bayesian Networks with Exaples in R*. Text in Statistical Science Series. Boca Raton, FL, USA: CRC Press, 2015.

Universidad Veracruzana