Introduction to Simulation
Reading: Law, Sections 1.1, 1.2, 1.8

Peter J. Haas

CS 590M: Simulation
Spring Semester 2020
Introduction to Simulation

Gambling Game
Definitions
More on Simulation
Key Issues in Simulation
Basic point estimates and confidence intervals
Discrete-Event Simulation
Course Goals
A Gambling Game

Is the following game a good bet over the long run?

- A fair coin is repeatedly flipped until $|\#\text{heads} - \#\text{tails}| = 3$
- Player receives $8.99$ at the end of the game but must pay $1$ for each coin flip

Approaches to answering the question:

- Try to compute the answer analytically (not easy)
- Play the game multiple times and use average reward to estimate expected reward (time-consuming)
- Use the power of the computer to experiment—Simulation!
Simulating coin flips on a computer: Pseudorandom numbers

- $U$ “looks like” a uniform random number between 0 and 1
- To generate:
  - Python: $U = \text{random.random}()$
  - C: $U = \text{(float)rand()} / \text{MAX\_RAND}$
  - Java: $U = \text{Math.random}()$
- Then “heads” if $0 \leq U \leq 0.5$ and “tails” if $0.5 < U \leq 1$

The need for careful simulation [Demo]

Simulation for science [NetLogo Demo]
Simulation: Definitions

Definition 1
A technique for studying real-world dynamical systems by imitating their behavior using a mathematical model of the system implemented on a digital computer.

Definition 2
A controlled statistical sampling technique for stochastic systems.

Q: Example of non-stochastic simulation?

Definition 3
A numerical technique for solving complicated probability models (analogous to numerical integration).
Monte Carlo methods

For static numerical problems

Example: Numerical integration with many dimensions
  ▶ WWII Manhattan Project: von Neumann, Teller, Turing

Will cover briefly in the course and homework
More on Simulation

Why simulation is awesome (mostly)

- Most frequently used tool of practitioners
- Interdisciplinary: spans Computer Science, Statistics, Probability, and Number Theory

Applications

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Biology (e.g., protein folding)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial risk</td>
<td>AI training</td>
</tr>
<tr>
<td>Video games</td>
<td>Flight simulation</td>
</tr>
<tr>
<td>Disease modeling</td>
<td>Astronomy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Safety</th>
<th>Military</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Telecom</td>
</tr>
<tr>
<td>Healthcare</td>
<td></td>
</tr>
</tbody>
</table>

Advantages and disadvantages

- Cheaper, faster, safer than dealing with real-world sys.
- Allows arbitrary model complexity
- Allows what-if analysis
- Can validate simpler models (analytic or simulation)
- Only gives approximate answers
- Can be expensive to create + costly to run (esp. if model is huge)
- Need deep system knowledge
- Subject to numerical issues
Simulation vs Machine Learning

Will the mechanism that generates data now generate it in the future?
(Not if I change the mechanism)

Allows What-If analyses

Extrapolation of 1970-2006 median U.S. housing prices

NCAR Community Atmosphere Model (CAM)

3.3 Eulerian Dynamical Core
\[
\begin{align*}
\frac{\partial \zeta}{\partial t} &= k \cdot \nabla \cdot \left( \frac{n}{\cos \phi} \right) + F_{\zeta W}, \\
\frac{\partial \zeta}{\partial \delta} &= \nabla \cdot \left( \frac{n}{\cos \phi} \right) - \nabla^2 \left( E + \Phi \right) + F_{\zeta H}, \\
\frac{\partial T}{\partial t} &= \frac{-1}{a \cos^2 \phi} \left[ \frac{\partial}{\partial \lambda} \left( UT + \cos \phi \frac{\partial}{\partial \phi} (VT) \right) + T \delta - \frac{\partial T}{\partial \eta} + \frac{R}{c_p} \frac{T v}{p} \right] + Q + F_{T H} + F_{F H}, \\
\frac{\partial q}{\partial t} &= \frac{-1}{a \cos^2 \phi} \left[ \frac{\partial}{\partial \lambda} \left( Uq + \cos \phi \frac{\partial}{\partial \phi} (Vq) \right) + q \delta - \frac{\partial q}{\partial \eta} + S, \\
\frac{\partial \pi}{\partial t} &= \int_1^\infty \nabla \cdot \left( \frac{\partial p}{\partial \eta} V \right) \, d\eta.
\end{align*}
\]
Simulation vs Machine Learning

Will the mechanism that generates data now generate it in the future?

Extrapolation of 1970-2006 median U.S. housing prices

NCAR Community Atmosphere Model (CAM)

3.3 Eulerian Dynamical Core
\[
\begin{align*}
\frac{\partial \zeta}{\partial t} &= k \cdot \nabla \times (n/\cos \phi) + F_{\zeta}, \\
\frac{\partial \delta}{\partial t} &= \nabla \cdot (n/\cos \phi) - \nabla^2 (E + \Phi) + F_{\delta}, \\
\frac{\partial T}{\partial t} &= \frac{-1}{a \cos^2 \phi} \left[ \frac{\partial}{\partial \lambda} (UT) + \cos \phi \frac{\partial}{\partial \phi} (VT) \right] + T\delta - \eta \frac{\partial T}{\partial \eta} + \frac{R}{c_p} \frac{T_v \omega}{p} \\
&+ Q + F_{T_H} + F_{T_H}, \\
\frac{\partial q}{\partial t} &= \frac{-1}{a \cos^2 \phi} \left[ \frac{\partial}{\partial \lambda} (Uq) + \cos \phi \frac{\partial}{\partial \phi} (Vq) \right] + q\delta - \eta \frac{\partial q}{\partial \eta} + S, \\
\frac{\partial \pi}{\partial t} &= \int_{1}^{\infty} \nabla \cdot \left( \frac{\partial p}{\partial \eta} \mathbf{V} \right) \, d\eta.
\end{align*}
\]
Simulation vs Machine Learning

Will the mechanism that generates data now generate it in the future? (Not if I change the mechanism)
Simulation vs Machine Learning

Will the mechanism that generates data now generate it in the future? (Not if I change the mechanism)

NCAR Community Atmosphere Model (CAM)

Allows What-If analyses
Simulation Resources

- **TOMACS**: ACM Transactions on Modeling and Computer Simulation
- **OR/MS Today** (biennial simulation software survey)
- **INFORMS Simulation Society**; see www.informs.org/Community/Simulation-Society
- **Winter Simulation Conference** proceedings; see http://informs-sim.org
  - Over 40 years of conference papers searchable by keyword
  - Introductory and advanced tutorials can be especially useful
- **Society for Computer Simulation**; see http://www.scs.org.
- **ACM SIGSIM**; see www.sigsim.org

See Sokolowski and Banks (Ch. 7) for extensive listing of simulation organizations and applications
Introduction to Simulation

Gambling Game
Definitions
More on Simulation

Key Issues in Simulation
Basic point estimates and confidence intervals
Discrete-Event Simulation
Course Goals
Overview of Simulation Process

Real-world system (existing or proposed) + Decision problem (Choose design or operating policy)

Mathematical simulation model
- states, events, clocks
- state transitions
- Input distributions
  - Probability theory
  - Fit distribution from data (maximum likelihood, Bayes)

Stochastic process definition
- \( \{X(t) : t \geq 0 \} \) or \( \{X_n : n \geq 0 \} \)
  - Discrete-time Markov chain (DTMC)
  - Continuous-time Markov chain (CTMC)
  - Semi-Markov process (SMP)
  - Generalized semi-Markov process (GSMP)

Sample path generation
- Uniform random numbers
- Non-uniform random numbers
  - Inversion, accept-reject, composition, convolution, alias method
  - Time-advance mechanism
  - Event list management

Output analysis
- Point estimates and confidence intervals
  - Simple means (SLLN and CLT based)
  - Nonlinear functions of means, quantiles (Taylor series, sectioning, jackknife, bootstrap)
  - Steady-state quantities: time-avg limits, delays (regenerative, batch means, jackknifing)

Efficiency improvement
- Common random numbers, antithetic variates, conditional Monte Carlo, control variates, importance sampling

Experimental design
- Factor screening
- Sensitivity analysis
- Metamodelling

Optimization
- Continuous (Robbins-Monro)
- Ranking and selection
- Discrete optimization
1. What questions are we trying to answer?
   - Complex, often dynamic (see Sawyer and Fuqua slides in Practitioner’s Gallery)
   - Identify stakeholders and available resources
   - Continual interplay with stakeholders during project
   - See also Conway & McClain
     http://pubsonline.informs.org/doi/pdf/10.1287/ited.3.3.13

2. How to model the system?
   - State definition, random variables, etc.
   - Operational vs policy models: different levels of detail
   - “As simple as possible” vs model re-use
Example of Model Formulation: Gambling game

Outcome of $i$th toss: $H_i = \begin{cases} 1 & \text{if } U_i \leq 0.5; \\ 0 & \text{if } U_i > 0.5 \end{cases}$

# of heads in first $n$ tosses: $S_n = \sum_{i=1}^{n} H_i$

# of tails in first $n$ tosses: $n - \sum_{i=1}^{n} H_i$

# heads - #tails: $2\sum_{i=1}^{n} H_i - n$

length of game: $L = \min \{ n \geq 1 : \left| 2\sum_{i=1}^{n} H_i - n \right| = 3 \}$

reward for game: $X = 9.99 - L$

Goal: estimate $\mu = E[X]$
3. Is the quantity that we are trying to estimate well defined?
   - Single-server queue with \( \rho > 1 \)
   - In gambling game, \( \mu \) defined iff \( P(L < \infty) = 1 \) and \( E[L] < \infty \)
   - Moral: do sanity checks!

4. How to generate run on a computer?
   - Gambling game is easy, industrial strength models are hard
   - In general, we will use low-level languages
     - Python, C/C++, Java versus Matlab, R
     - For deep understanding of foundational principles
     - Flexibility, low cost, fast execution
     - Programming ability strengthens your resume
5. How do we verify the simulation?

- **Verification**: Correctness of the computer implementation of the simulation model

- **Good coding practices:**
  - make debugging easy (e.g. use print statements)
  - write modular code (and unit-test it)
  - Lots of comments
  - Avoid too many global variables
6. How do we validate the simulation?

- **Validation**: Adequacy of the simulation model in capturing system of interest
- Beware of over-fitting: use, e.g., cross validation [Hastie et al., *Elements of Statistical Learning*, Sec. 7.10]
- Beware that good fit to current data $\not\Rightarrow$ good extrapolation
- Aim for *insights*: trends and comparisons
- Use *sensitivity analysis* to build credibility
Key Issues, Continued

7. Number and length of simulation runs?

8. Can the simulation be made more efficient?
   - Statistical and computational efficiency

9. How do we use simulation to make decisions?
   - Compare systems: ranking and selection
   - Set operating or design parameters: stochastic optimization
   - Set operating policies: reinforcement learning, Markov decision processes
Introduction to Simulation

Gambling Game
Definitions
More on Simulation
Key Issues in Simulation

Basic point estimates and confidence intervals
Discrete-Event Simulation
Course Goals
Point Estimates & Strong Law of Large Numbers

Estimating expected reward in gambling game

- Replicate experiment (i.e., play game) \( n \) times to get \( X_1, X_2, \ldots, X_n \)
- Estimate expected reward by \( \mu_n = \frac{1}{n} \sum_{i=1}^{n} X_i \)
- Why is this a reasonable estimate?

Strong law of large numbers

- Suppose \( X_1, X_2, \ldots \) are i.i.d. with finite mean \( \mu \)
- Then, with probability 1,

\[
\frac{1}{n} \sum_{i=1}^{n} X_i \to \mu \text{ as } n \to \infty
\]
Confidence Intervals & Central Limit Theorem

How do we assess the error in our estimate?

- Need to distinguish true system differences from random fluctuations

\[ \frac{\sqrt{n}}{\sigma} (\bar{X}_n - \mu) \stackrel{D}{\approx} N(0,1) \rightarrow \mu_n - \mu \stackrel{D}{\approx} N(0, \frac{\sigma^2}{n}) \]

Central Limit Theorem

- Suppose \( X_1, X_2, \ldots \) are i.i.d., mean \( \mu < \infty \) and variance \( \sigma^2 < \infty \)
- Then

\[ \frac{\sqrt{n}}{\sigma} \left( \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right) \Rightarrow N(0,1) \]

as \( n \to \infty \), where \( N(0,1) \) is a standard normal random variable and \( \Rightarrow \) denotes convergence in distribution

- Intuitively, the sample average \( \mu_n \) is approximately distributed as \( N(\mu, \sigma^2/n) \) when \( n \) is large (\( \geq 50 \))
Confidence Interval for Fixed Sample Size

To compute $100(1 - \delta)\%$ confidence interval:

- Choose $z_\delta$ such that $P(-z_\delta \leq N(0, 1) \leq z_\delta) = 1 - \delta$
- Equivalently, $P(N(0, 1) \leq z_\delta) = 1 - \delta/2$
- Can find in Table T1 (p. 716) in the textbook

- By CLT,
  
  $$P \left\{ -z_\delta \leq \frac{\sqrt{n}(\mu_n - \mu)}{\sigma} \leq z_\delta \right\} \approx 1 - \delta$$
  
  or, after algebra,
  
  $$P \left\{ \mu_n - \frac{z_\delta \sigma}{\sqrt{n}} \leq \mu \leq \mu_n + \frac{z_\delta \sigma}{\sqrt{n}} \right\} \approx 1 - \delta$$
  
  so random interval

  $$\left[ \mu_n - \frac{z_\delta \sigma}{\sqrt{n}}, \mu_n + \frac{z_\delta \sigma}{\sqrt{n}} \right]$$

  covers true value with probability $\approx 1 - \delta$
Problem: $\sigma^2$ is unknown

► Solution: Estimate $\sigma^2$ from data: 

$$s_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \mu_n)^2$$

Final 100$(1 - \delta)$% CI formula:

$$\left[ \mu_n - \frac{z_\delta s_n}{\sqrt{n}}, \quad \mu_n + \frac{z_\delta s_n}{\sqrt{n}} \right]$$

The quantity $z_\delta s_n/\sqrt{n}$ is called the half-width of the CI

Questions:

► How, roughly, do I cut my error in half?
► What can go wrong if $n$ is too small?
Choosing the Number of Simulation Runs

### Trial runs
- Generate $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_k$ (where $k \geq 50$)
- Compute $\hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} \hat{X}_i$ and $\hat{s}^2 = \frac{1}{k-1} \sum_{i=1}^{k} (\hat{X}_i - \hat{\mu})^2$
- **Absolute** precision intervals
  - Estimate $\mu$ to within $\pm \varepsilon$ with probability $100(1 - \delta)\%$
  - Want to choose $n$ so that $\frac{\sigma Z_\delta}{\sqrt{n}} = \varepsilon$: $n = \frac{\hat{s}^2 Z_\delta^2}{\varepsilon^2}$
- **Relative** precision intervals
  - Estimate $\mu$ to within $\pm 100\varepsilon\%$ with probability $100(1 - \delta)\%$
  - Want to choose $n$ so that $\frac{\sigma Z_\delta}{\sqrt{n}} = \varepsilon \mu$: $n = \frac{\hat{s}^2 Z_\delta^2}{\varepsilon^2 \hat{\mu}^2}$

### Sequential estimation
- Simulate until interval is narrow enough
- Asymp. valid as $\varepsilon \to 0$ [Nadas, *Ann. Math Statist.*.,1969]
- Danger: premature stopping