ORIE 4580/5580: Simulation Modeling and Analysis

ORIE 5581: Monte Carlo Simulation

Fall 2025

Sid Banerjee

School of ORIE, Cornell University

class poll 1

4580 / 5580 / 5581

PollEverywhere: https://pollev.com/simulationf25

two teams, A and B, play a best of ~~5~~ series, starting with a home game for A, and then alternating home games till there is a winner (i.e., till one team wins 3 games)

- $\mathbb{P}$[team wins home game] = 0.51 ← 'home bias'
- $\mathbb{P}$[team wins away game] = 0.49

which team has a higher probability of winning the series?

(a) A 86%

(b) B 9%

(c) both are equal

different orders

ABABA – AAA

AB →

BBAAA

AABBB

class poll 2

you have two coins, red and green

- $\mathbb{P}$[heads on green coin] $= 0.51$ ← 'slight bias towards H'
- $\mathbb{P}$[heads on red coin] $= 0.49$ ← ' " " " T'

suppose you toss the red coin 2 times and the green coin 3 times

which of the following is more likely?

92%

(a) you see more heads

$$\mathbb{P}[\underbrace{\#\text{ of H} > \#\text{ of T}}_{\text{'more heads'}}] > \mathbb{P}[\underbrace{\#\text{ of T} > \#\text{ of H}}_{\text{'more tails'}}]$$

≈ 4%

(b) you see more tails

$$\mathbb{P}[\underbrace{\#\text{ of H} > \#\text{ of T}}_{\text{'more heads'}}] < \mathbb{P}[\underbrace{\#\text{ of T} > \#\text{ of H}}_{\text{'more tails'}}]$$

≈ 4%

(c) both are equally likely

class poll: solution

two teams, A and B, play a best of 7 series, where the first 3 games are home games for A, and the last 4 are home games for B; the first to win 4 games wins the series

- $\mathbb{P}$[team wins home game] $= 0.51$
- $\mathbb{P}$[team wins away game] $= 0.49$

which team has a higher probability of winning the series?

B more likely to win

- Identical to question 2

Important - events are independent

- Can 'simulate' ALL games, and then see who wins

HTHTT = ABABB ⇒ B wins

HHHTT = AAA $\boxed{BB}$ ⇒ A wins

not played

⋮

simulation = probability + computing

Programming with a random number generator

- understanding what stochastic model do / don't do
- 'power of randomness' in computing

'what if' expts

Eg - computing integrals

why study simulation?

- develop probabilistic thinking
- understand how to summarize and present data - Simulation data is 'random'
- learn how to make stochastic models of complex systems - 2nd half
- tool for fast computation and algorithms - Monte Carlo methods

applications

first half - all 3 sections

Monte Carlo methods

probability in computing has five major applications

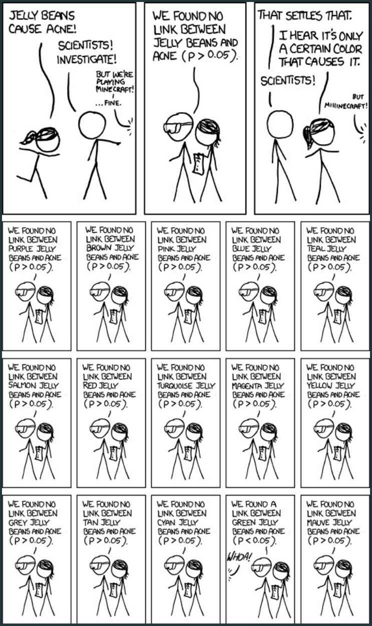
- **numerical computation**: Monte Carlo algorithms for scientific computing
- **algorithms for massive data**: sampling, sketching, streaming data, random-walk network algorithms, graphical models, etc.
- risk analysis: quantifying and hedging against random 'shocks' in daily life
- counterfactual ('what-if') analysis: understanding and optimizing complex systems in-vitro
- **cryptography, privacy and game theory**: secret keys, fingerprinting, differential privacy, mixed strategies

Stochastic modeling

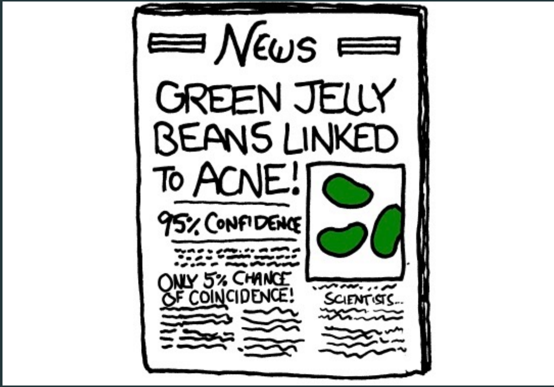
2nd half - discrete event simulation (ie, Markov Chain models)
- only for 4580/5580
- continuous time/space models - Brownian motion (5581 → 5582)

understanding randomness and sampling

xkcd.com



'p-hacking'



what is the misunderstanding here?

- how do we interpret data from random trials?

risk analysis: choosing projects

- current debate - what is a 'tolerable' probability that AGI destroys the world?

what are the fundamental misunderstandings here?

- the project has a 70% chance of success. Why not try?
- why not increase the chance of a success by doing 10?

counterfactual analysis ('what-if' analysis)

COVID-19 Mathematical Modeling for Cornell's Fall Semester

PhD Students: J. Massey Cashore, Ning Duan, Alyf Janmohamed, Jiayue Wan, Yujia Zhang
Faculty: Shane Henderson, David Shmoys, Peter Frazier*

June 15, 2020

Executive Summary:

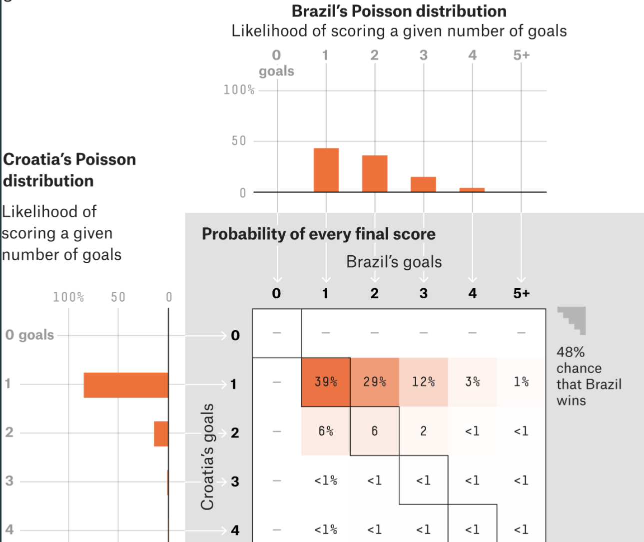
- Initial modeling results suggest that a combination of contact tracing, asymptomatic surveillance, and low initial prevalence (supported through testing students prior to, and upon, returning to campus) can achieve meaningful control over outbreaks on Cornell's Ithaca campus in the fall semester if asymptomatic surveillance is sufficiently frequent and if we have sufficient quarantine capacity. This would dovetail with a complementary effort at Cornell to reduce transmissions through housing policy, class organization, and regulations on social gatherings.

- We use our model to predict outcomes for a full return of students, faculty and staff in the fall semester over a 16 week time period, with cases imported from returning students and from Tompkins county, counterbalanced by aggressive asymptomatic surveillance where every member of the campus community is tested every 5 days. The course of the epidemic is random and we directly model that randomness. Accordingly, our model produces a range of potential futures. In the median random potential future, under our nominal set of parameters, 3.6% of the campus population (1254 people) become infected, and 0.047% of the campus population (16 people) require hospitalization. The 90% quantile rises to 4.02% infected and 0.051% requiring hospitalization. Of the 1254 infections in the median outcome, 570 are due to direct

prediction

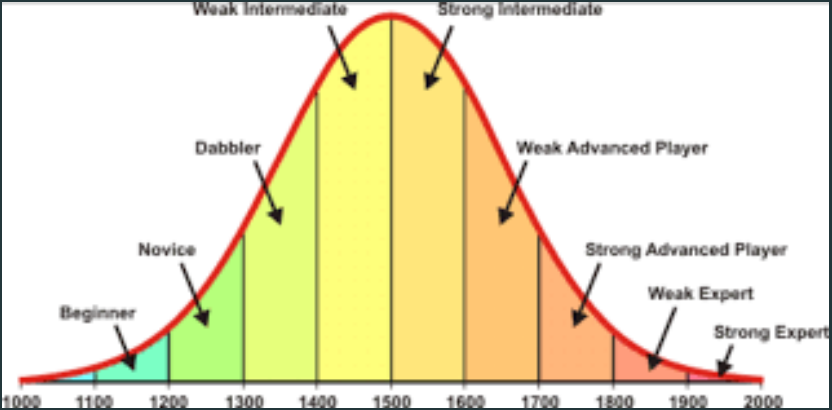
Forecasting a live match

Brazil vs. Croatia, June 12, 2014, in the 65th minute with the game tied 1-1



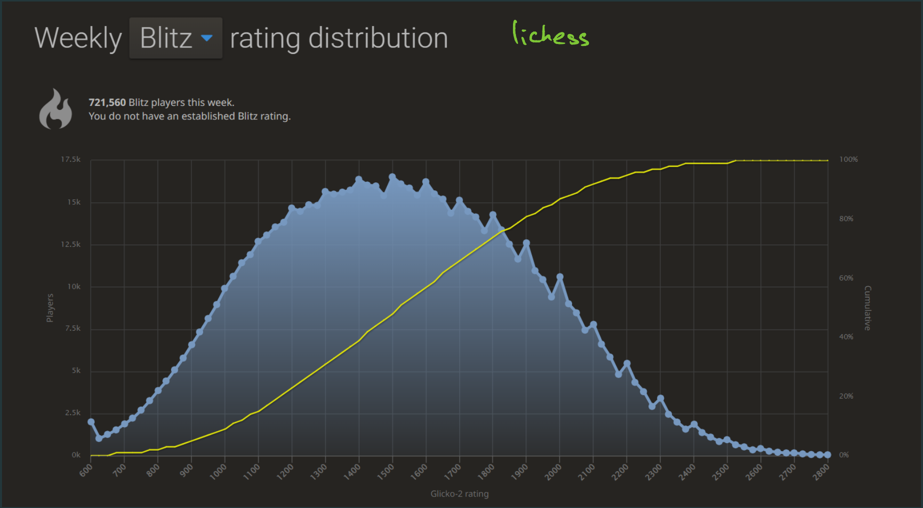
	0	1	2	3	4	5+
0	–	–	–	–	–	–
1	–	39%	29%	12%	3%	1%
2	–	6%	6	2	<1	<1
3	–	<1%	<1	<1	<1	<1
4	–	<1%	<1	<1	<1	<1

quantifying quality

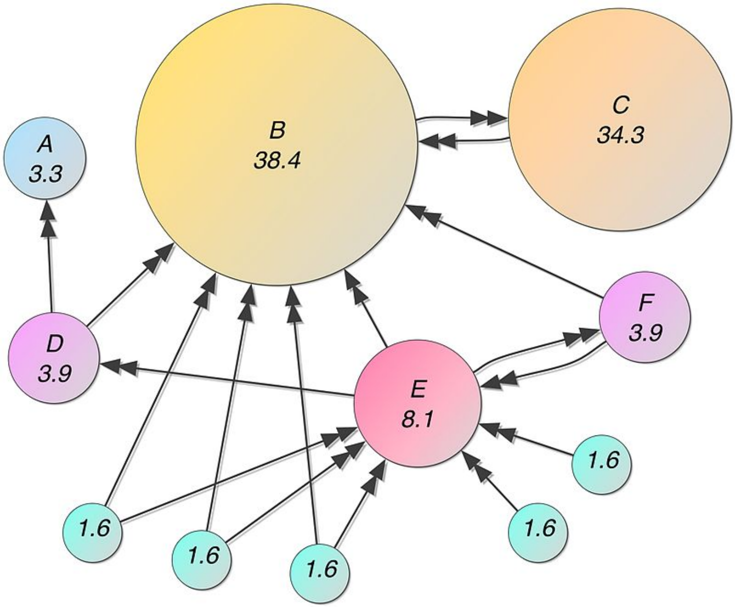


for example Elo ratings, xG/xP

quantifying quality ('real' ELO ratings)



quantifying utility (wikipedia example for PageRank)



. . .and beyond: optimization, control and RL

README.md

OR Suite

Reinforcement learning (RL) is a natural model for problems involving real-time sequential decision making. In these models, a principal interacts with a system having stochastic transitions and rewards and aims to control the system online (by exploring available actions using real-time feedback) or offline (by exploiting known properties of the system).

These project revolves around providing a unified landscape on scaling reinforcement learning algorithms to operations research domains.

Installation Guide

In order to install the required dpeendencies for a new conda environment, please run:

```
conda create --name ORSuite python=3.8.5
conda activate ORSuite
python -m pip install -r requirements.txt
```

High-Level Overview

The repository has three main components as a traditional Reinforcement Learning set-up :

1. Environments : Environment for the agent to interact with and reside in. `~/or_suite/envs`
2. Agents : Choice of Algorithm `~/or_suite/agents`
3. Experiments : This is a take on implementing the enviroment and agents with a choice of algorithm `~/or_suite/experiment`

Contributors 6

Languages

- Jupyter Notebook 66.7%
- Python 33.3%

ORSuite: Benchmarking Suite for Sequential Operations Models

Christopher Archer Siddhartha Banerjee Mayleen Cortez
Carrie Rucker Sean R. Sinclair* Max Solberg
Qiaomin Xie Christina Lee Yu

ABSTRACT

Reinforcement learning (RL) has received widespread attention across multiple communities, but the experiments have focused primarily on large-scale game playing and robotics tasks. In this paper we introduce *ORSuite*, an open-source library containing environments, algorithms, and instrumentation for operational problems. Our package is designed to motivate researchers in the reinforcement learning community to develop and evaluate algorithms on operational tasks, and to consider the true *multi-objective* nature of these problems by considering metrics beyond cumulative reward. [1]

1. INTRODUCTION

Reinforcement learning (RL) is a natural model for problems involving real-time sequential decision making, includ-

or the efficacy in a resource allocation problem. Moreover, many of the problems naturally have continuous or combinatorial state and action spaces, which makes designing RL algorithms for these domains an important direction for the research community.

1.1 Our Contributions

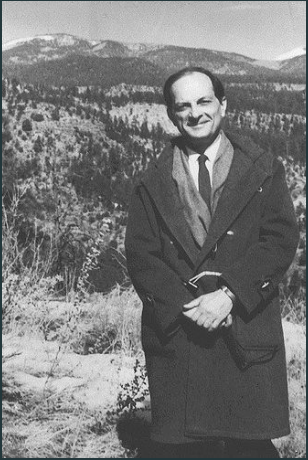
In this note we outline the structure and purpose of the *ORSuite* package [21]. Our goal is to provide a standardized library for the reinforcement learning community to explore applying algorithms to problems arising in operations research. The package is aimed at modeling common operational problems, providing implementation of existing heuristic approaches to these problems, and optional instrumentation to benchmark the performance of algorithms

some history and demos

birth of Monte Carlo Simulation: Ulam in hospital

"...in 1946 as I was convalescing from an illness and playing solitaires ... I thought what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics..."

Monte Carlo



Stanislaw Ulam

Monte Carlo simulation at Los Alamos

progress in simulation was driven in the early years by the development of the MANIAC computer at Los Alamos National Labs by John von Neumann and Nicholas Metropolis

John von Neumann

Nicholas Metropolis

MCMC - Metropolis method

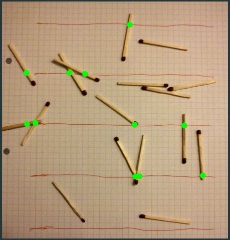
an even older simulation!

most famous early example of simulation: Buffon's needle problem (18th century)

- throw matches of length 1 on horizontal grid with parallel lines at distance 1
- compute $\hat{X} = 2 \times D/C$, where

  D =number dropped

  C =number which touch line



Comte de Buffon

see Numberphile video for more details

logistics

Essential Course Information (contd.)

- **lectures**
  time: TR 11:40am-12:55pm
  location: Kimball Hall B11
- **recitation sessions**
  1. Monday 2:30pm–4:25pm, Rhodes 571
  2. Monday 7:30pm–9:25pm, Rhodes 571
  3. Wednesday 7:30pm–9:25pm, Rhodes 571
- **course communication**:
  - **Canvas**: https://canvas.cornell.edu/courses/80773
  - **Gradescope**: https://www.gradescope.com/courses/1080743
  - **Ed Discussion**: https://edstem.org/us/courses/81617

All class communications on Ed

course resources

- **course notes**: blank slides uploaded on Canvas before class – with annotations after class
- **PollEverywhere**: `https://pollev.com/simulationf25`
- **software**
  - all coding in Python + Jupyter notebook
  - we recommend Google Colab for collaborative coding

homeworks

weekly homework assignments (5 for ORIE 5581, 8 for 4580/5580)

- solutions must be submitted online on Gradescope
  students must *typeset* all solutions
- homeworks due on Thursday at 11.59pm (midnight)
- **collaboration**: should do homework in pairs; submit a single solution with both names and netids on the solution
- **late submissions and drops**:
  3 late days for 4580/5580
  2 late days for 5581 across all homeworks (at most 1 late days per hw)
  all students can drop 1 lowest homework grade

grading

- **exams**:
  - **prelim 1**: October 16th, 7.30pm: for 4580/5580/5581
  - **prelim 2**: December 2nd, in class: for 4580/5580
  - **no final exam**

- (tentative) grading scheme

component	4580/5580	5581
class participation	5	5
homeworks	30	45
project	30	-
exam(s)	35	55

- class participation: mainly via online polls (but also recitations, Ed discussion)

course outline

topics cover simulation analysis, modeling and optimization

review of probability	1-2
basic Monte Carlo Simulation	1-2
generating random variables	2
input modeling	3
variance reduction and importance sampling	2-3
intro to Markov Chains	1
intro to discrete-event system simulation	1
Markovian models and queues	2-3
output analysis	2
comparing systems; ranking and selection	1-2
optimization, control and reinforcement learning	2-3
Markov chain Monte Carlo	2