CS7480
Topics in Programming Languages: Probabilistic Programming

Instructor: Steven Holtzen

Place: Northeastern University

Term: Fall 2021

Course webpage:
https://www.khoury.northeastern.edu/home/sholtzen/CS7480Fall21/

Overview

1. Course logistics and introduction
2. Overview of course content and format
3. Tiny technical intro: what are probabilistic programs?

What is this course?

- A graduate-level seminar on probabilistic programming languages (PPL)
- Goals for this class (what you will know by the end)
  - In short: ***bringing you up to speed on the modern research landscape of probabilistic programming***
  - What PPLs are and what they are useful for
  - How to design and implement your own PPL
  - How to read probabilistic programming papers
  - How to design and execute a research project using PPLs
  - …
- An experimental new course! We will learn as we go

Who is the audience?

- PhD. and advanced Master’s students from programming languages, AI, or machine learning
- Expected background:
  - Ability to program in a common programming language
  - Familiarity (or willing to learn) some mathematical foundations from logic, computational complexity, and probability
  - Ability to write a technical paper
- Talk to Steven if you have any concerns/questions

COVID-19

- The class will be in-person by default (for now)
  - Everyone is required to always wear a mask in the classroom.
  - No eating or drinking in the classroom please; feel free to excuse yourself for these activities.
  - You *are required* to wear the mask correctly.
  - Follow CDC mask guidelines https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/about-face-coverings.html
  - This is for the comfort and safety of the instructor and your fellow students; please be respectful.
- If you ***ever*** feel uncomfortable with any of these aspects of the course, talk to Steven

Course Resources

- **Course webpage** (main source for content, “ground truth”): https://www.khoury.northeastern.edu/home/sholtzen/CS7480Fall21/
- **Canvas**: turn things in, get private links
  - All slides I make will be available on canvas
- **Office Hours**:
  - Tuesday 11AM – 12PM in 310D WVH
  - Also by appointment on Zoom (email me)
- **Email me** (please include “7480” in the subject) s.holtzen@northeastern.edu

Personal/Interpersonal Resources

- I aim for the class to be an *inclusive environment* where ***anyone*** feels empowered to share their point of view
  - This is a research class: do not be afraid of speaking up or volunteering
- Having trouble in the class? Feeling stressed?
  - Email me or come to my office hours: we can work something out and I can help you find resources
  - Mental health resources: Find@Northeastern https://www.northeastern.edu/uhcs/find-at-northeastern/

Introductions

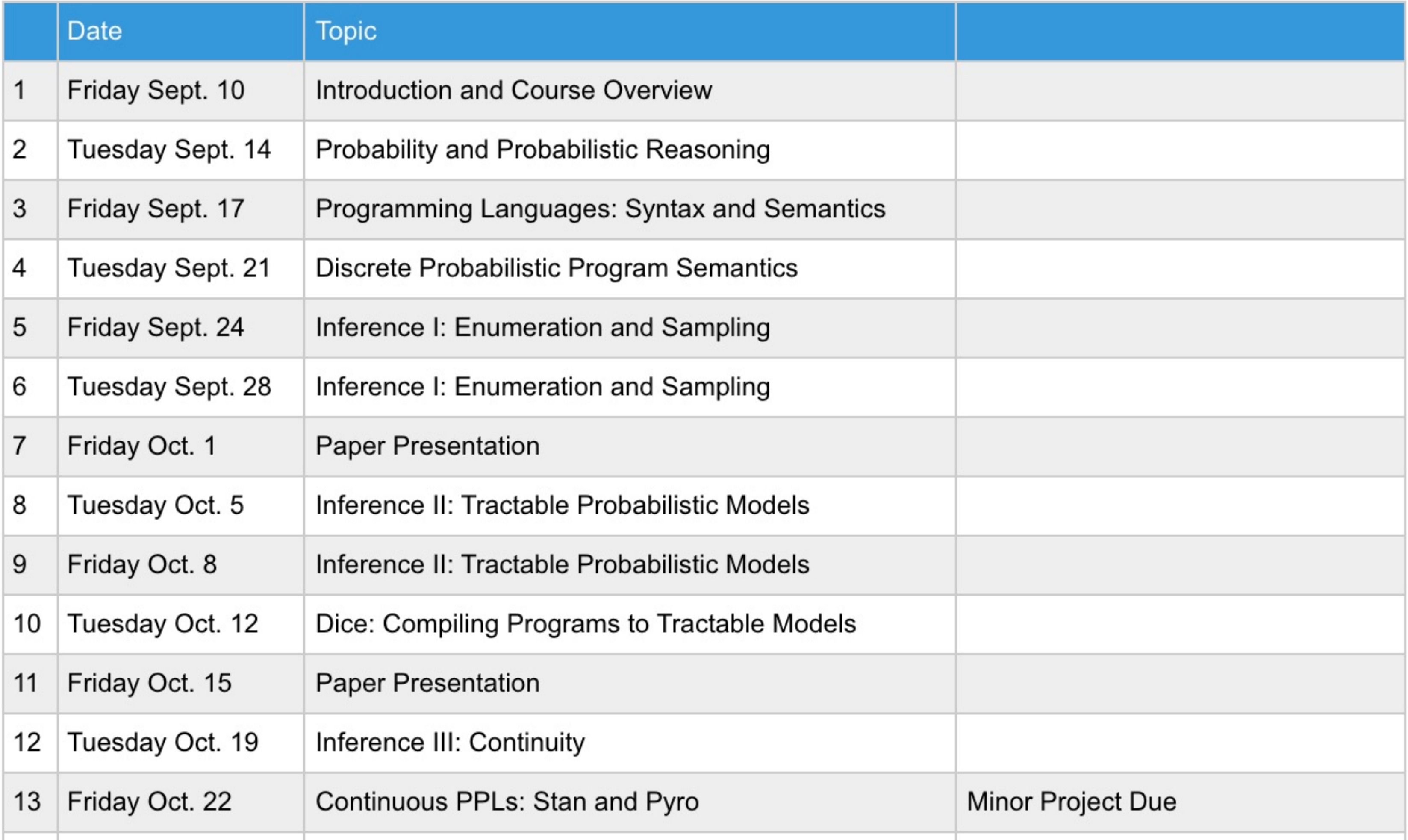
- Let's take a moment to get to know everyone else in the class
- Me:

- PhD. from UCLA in June 2021
- Grew up in LA
- Got my PhD. in probabilistic programming language (and related topics)
- Looking to work with students on research! Part of why I am teaching this class.

Course Overview

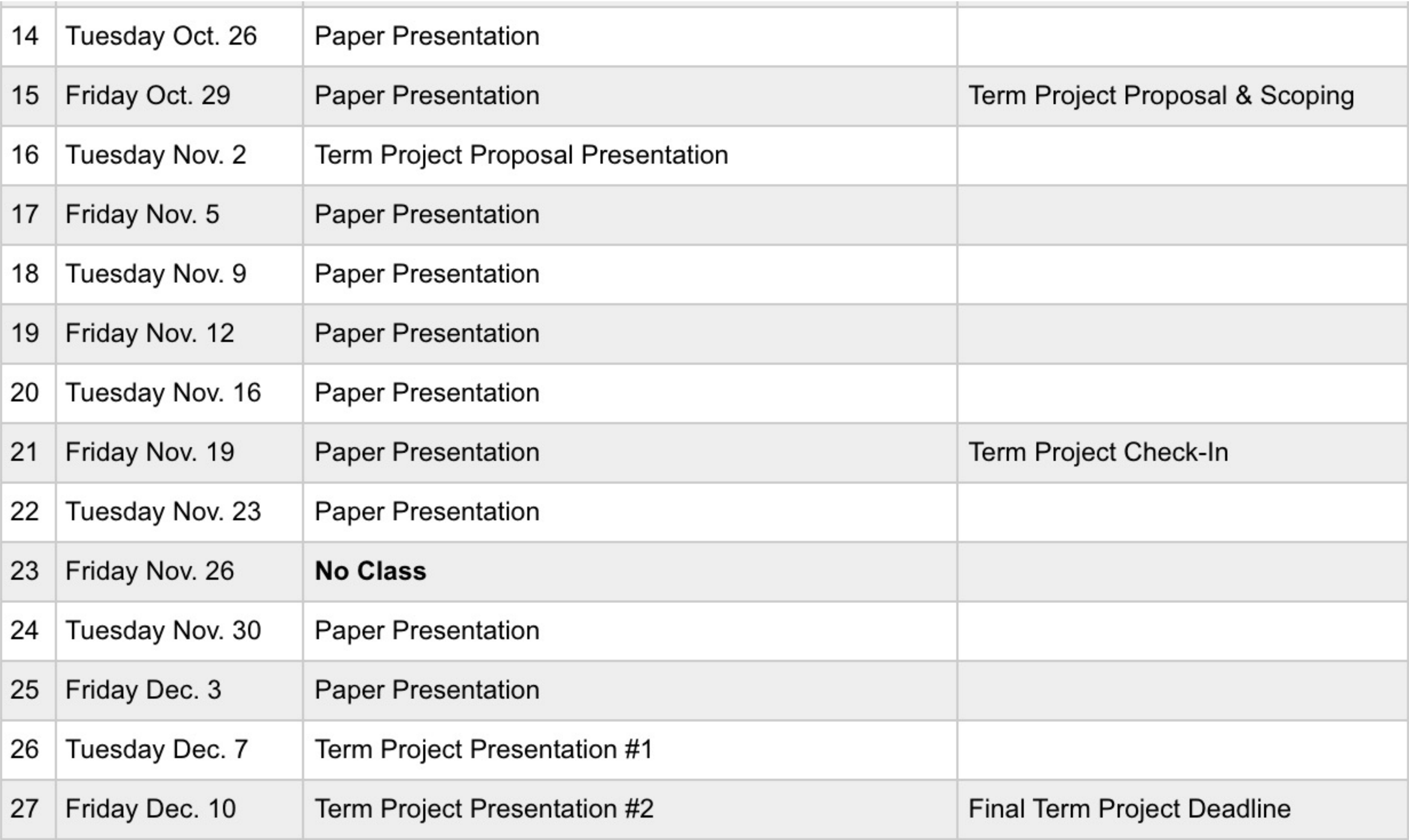
- Probabilistic programming is a new discipline
  - Though it has gone through a few births/rebirths…
- Courses on the topic will vary heavily in how they are taught based on instructor preferences ☺

Course Overview



	Date	Topic	
1	Friday Sept. 10	Introduction and Course Overview	
2	Tuesday Sept. 14	Probability and Probabilistic Reasoning	
3	Friday Sept. 17	Programming Languages: Syntax and Semantics	
4	Tuesday Sept. 21	Discrete Probabilistic Program Semantics	
5	Friday Sept. 24	Inference I: Enumeration and Sampling	
6	Tuesday Sept. 28	Inference I: Enumeration and Sampling	
7	Friday Oct. 1	Paper Presentation	
8	Tuesday Oct. 5	Inference II: Tractable Probabilistic Models	
9	Friday Oct. 8	Inference II: Tractable Probabilistic Models	
10	Tuesday Oct. 12	Dice: Compiling Programs to Tractable Models	
11	Friday Oct. 15	Paper Presentation	
12	Tuesday Oct. 19	Inference III: Continuity	
13	Friday Oct. 22	Continuous PPLs: Stan and Pyro	Minor Project Due

Course Overview



14	Tuesday Oct. 26	Paper Presentation	
15	Friday Oct. 29	Paper Presentation	Term Project Proposal & Scoping
16	Tuesday Nov. 2	Term Project Proposal Presentation	
17	Friday Nov. 5	Paper Presentation	
18	Tuesday Nov. 9	Paper Presentation	
19	Friday Nov. 12	Paper Presentation	
20	Tuesday Nov. 16	Paper Presentation	
21	Friday Nov. 19	Paper Presentation	Term Project Check-In
22	Tuesday Nov. 23	Paper Presentation	
23	Friday Nov. 26	**No Class**	
24	Tuesday Nov. 30	Paper Presentation	
25	Friday Dec. 3	Paper Presentation	
26	Tuesday Dec. 7	Term Project Presentation #1	
27	Friday Dec. 10	Term Project Presentation #2	Final Term Project Deadline

Paper Presentations

- Goal: learn how to navigate the modern literature on probabilistic programming
  - There is no textbook for this content (yet)! Learning it really requires reading papers
- Discussant chooses a paper
  - Fills out questionnaire
- *Everyone* (even me!) fills out the reader's questionnaire before lecture

Evaluation

- Standard letter grading
- 20%: Course Participation and Presentations
  - Participation is an important part of this class. Please let Steven know if you need to miss class; no excuse is necessary.
- 30%: Minor Project (Link to syllabus)
  - Implementing a simple PPL
- 50%: Final Project (Link to syllabus)
  - A more research-oriented term project

Minor Project: SimPPL

- Implement a **sim**ple **p**robabilistic **p**rogramming **l**anguage
- I will give you the language specification
- A draft of the project description is already online
- Implement two inference algorithms
- (Tentatively) Due Oct 22
- Will discuss more details about this later in class

Term Project

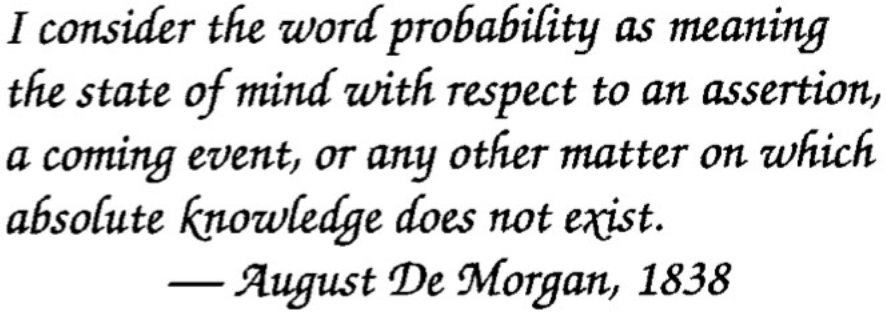
- Self-directed research project due at the end of the term
- May work in up to teams of 2, not required
- You choose the topic and define your own metrics of success (I will help you do this)
- Syllabus is online; feel free to read
- Goal: deeply engage with the material, try something new, implement something from a paper, extend `SimPPL` in some interesting way, …

Course Success Tips

- Work with Steven and your fellow students
  - Office hours, email, find a buddy
- Start projects early: give yourself time to get stuck and find help
- Don’t be surprised if there is something you don’t know; it’s an advanced seminar with many backgrounds
  - Ask questions during class
  - Search online
  - Come to office hours

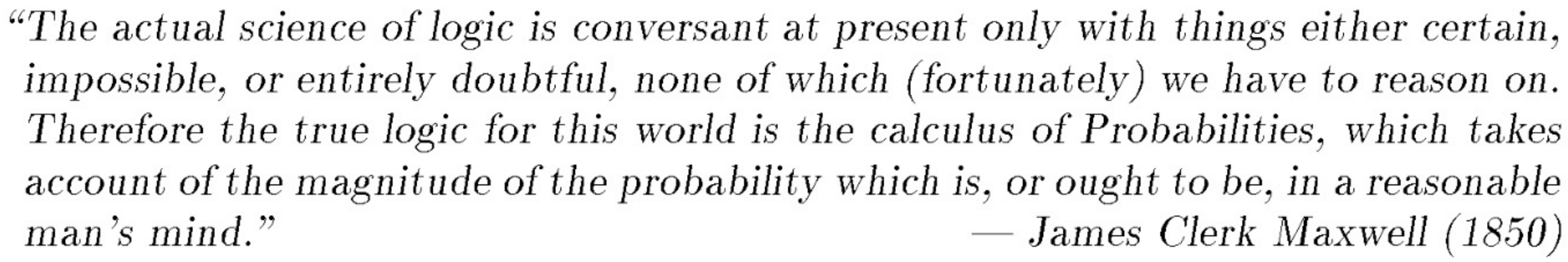
probabilistic programing

What is Probability?



I consider the word probability as meaning the state of mind with respect to an assertion, a coming event, or any other matter on which absolute knowledge does not exist.

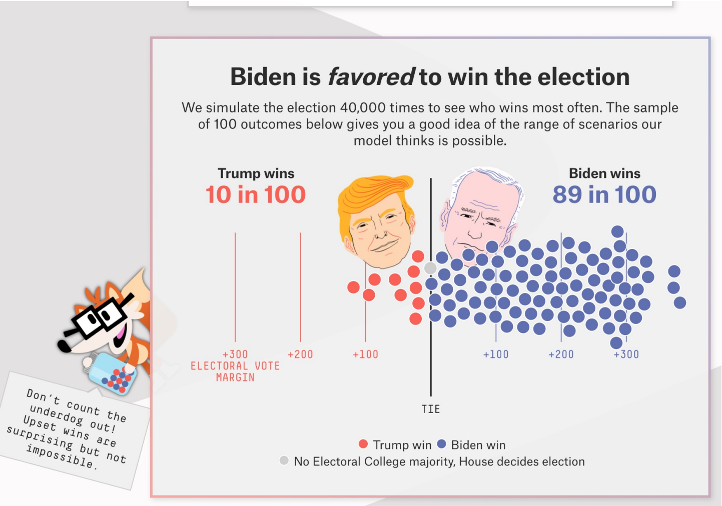
— August De Morgan, 1838



"The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind."

— James Clerk Maxwell (1850)

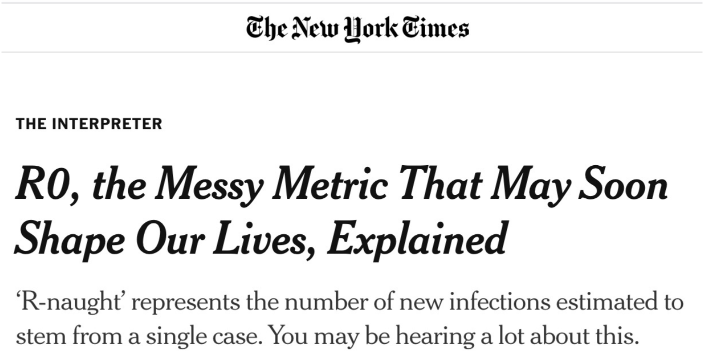
Probability in Our Lives



Biden is *favored* to win the election

We simulate the election 40,000 times to see who wins most often. The sample of 100 outcomes below gives you a good idea of the range of scenarios our model thinks is possible.

Don't count the underdog out! Upset wins are surprising but not impossible.

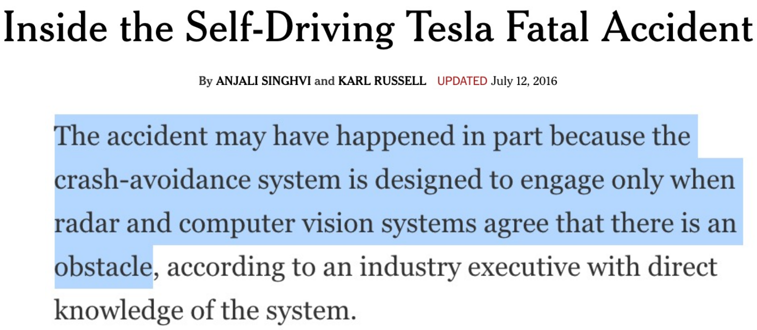


The New York Times

THE INTERPRETER

R0, the Messy Metric That May Soon Shape Our Lives, Explained

‘R-naught’ represents the number of new infections estimated to stem from a single case. You may be hearing a lot about this.

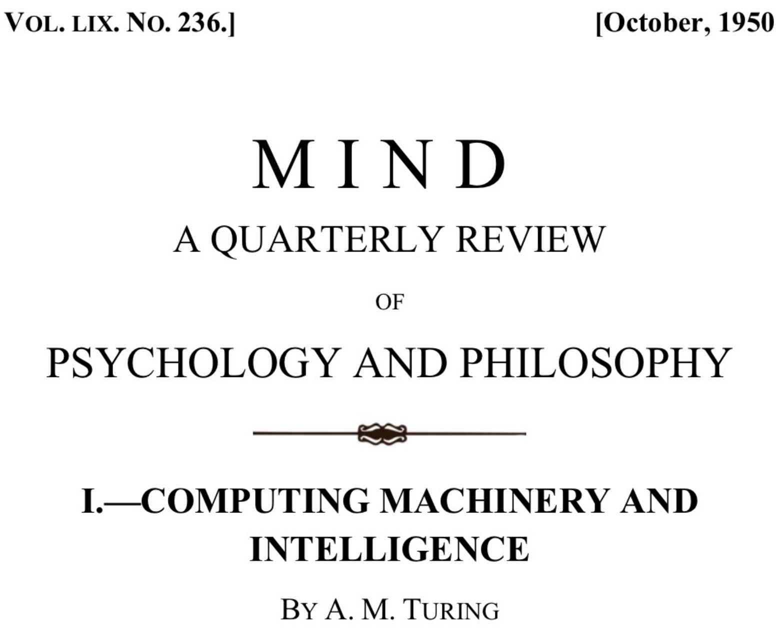


Inside the Self-Driving Tesla Fatal Accident

By ANJALI SINGHVI and KARL RUSSELL UPDATED July 12, 2016

The accident may have happened in part because the crash-avoidance system is designed to engage only when radar and computer vision systems agree that there is an obstacle, according to an industry executive with direct knowledge of the system.

Probability in AI and Computation



VOL. LIX. NO. 236.] [October, 1950

M I N D

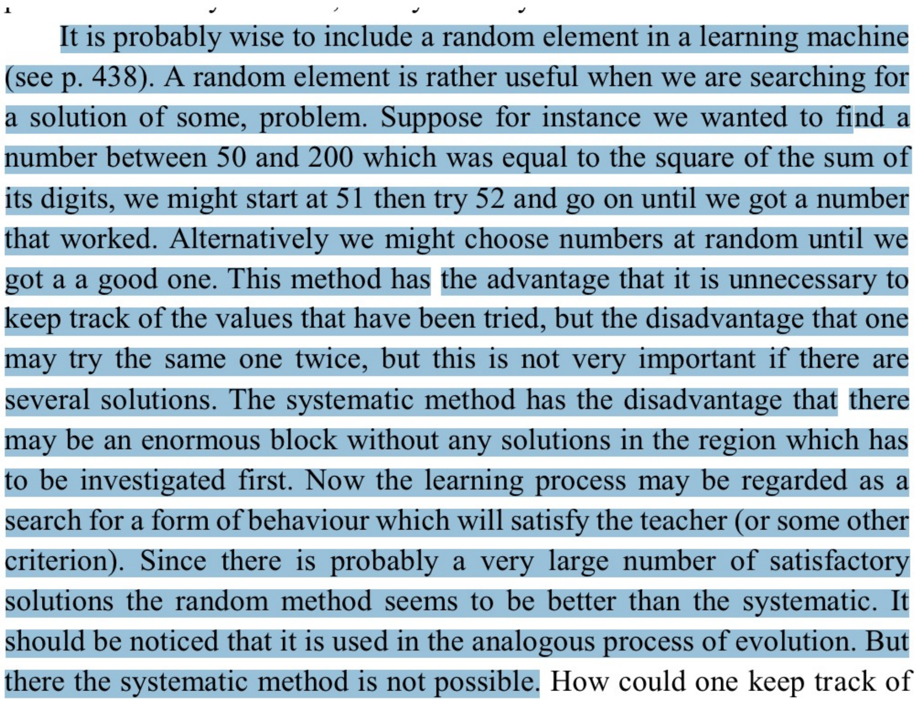
A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING



It is probably wise to include a random element in a learning machine (see p. 438). A random element is rather useful when we are searching for a solution of some, problem. Suppose for instance we wanted to find a number between 50 and 200 which was equal to the square of the sum of its digits, we might start at 51 then try 52 and go on until we got a number that worked. Alternatively we might choose numbers at random until we got a a good one. This method has the advantage that it is unnecessary to keep track of the values that have been tried, but the disadvantage that one may try the same one twice, but this is not very important if there are several solutions. The systematic method has the disadvantage that there may be an enormous block without any solutions in the region which has to be investigated first. Now the learning process may be regarded as a search for a form of behaviour which will satisfy the teacher (or some other criterion). Since there is probably a very large number of satisfactory solutions the random method seems to be better than the systematic. It should be noticed that it is used in the analogous process of evolution. But there the systematic method is not possible. How could one keep track of

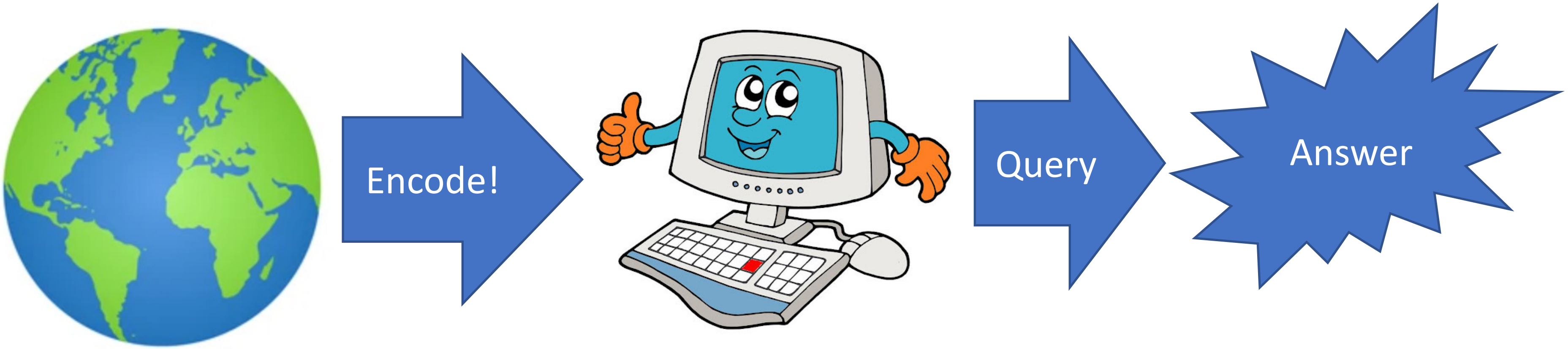
Probabilistic modeling

- Probability is *important*…
  - *But*, it's hard to do it yourself

- The world is complicated

Probabilistic modeling

- *We need computer systems for helping us use probabilities in our daily lives*



Encode!

Query

Answer

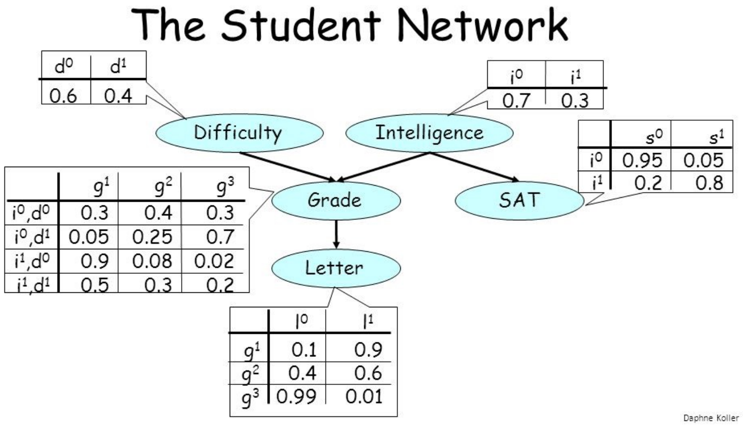
- How we *encode* and how we *query* are the main subjects of the course
  - Need *encoding and query languages*

Probabilistic graphical models

J. Pearl

- Choice of encoding language is *very important*

Encode!



The Student Network

d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

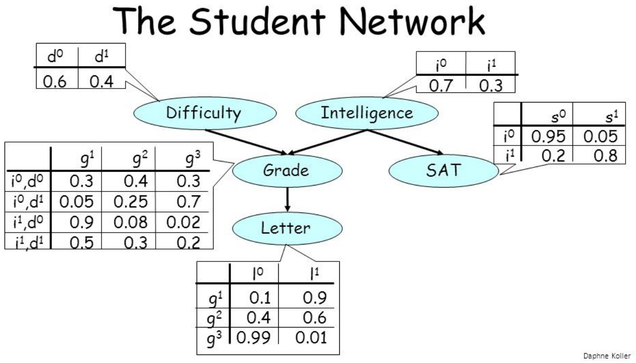
Daphne Koller

- Represent relationships between random variables as a *directed graph*

Probabilistic graphical models

- Graphs as a language for specifying probability distributions are *not ideal…*

- Not a very intuitive language…



The Student Network

d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Daphne Koller

Probabilistic Programming Languages

- Use the syntax of a *programming language* to make a probabilistic model

```
patient.flu = flip 0.001
if(patient.flu) {
   patient.cough = flip(0.9)
   patient.temp = gaussian(101, 3)
} else {
…
```

- Goal: **Inference**, compute the probability that the program outputs a particular value

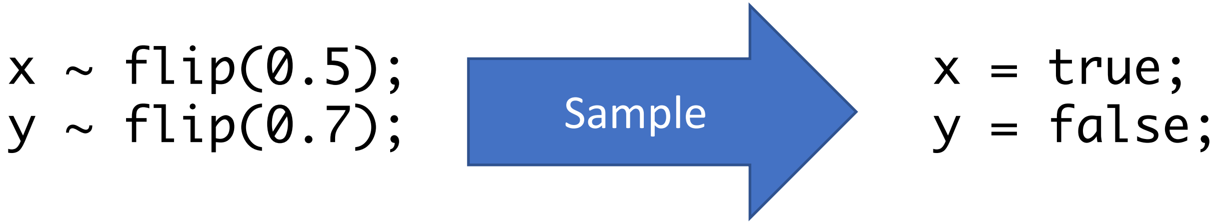
Your First Probabilistic Program

- It looks like this:

```
x ~ flip(0.5);
y ~ flip(0.7);
```

`x ~ flip(θ);` means “flip a coin, and output true with probability θ

- So we could “run” this program by evaluating each flip, then executing the program:



```
x ~ flip(0.5);
y ~ flip(0.7);
```

```
x = true;
y = false;
```

Your First Probabilistic Program

- Possible worlds for

```
x ~ flip(0.5);
y ~ flip(0.7);
```

```
x = true;
y = true;
```

ω_1

```
x = false;
y = true;
```

ω_2

```
x = false;
y = false;
```

ω_3

```
x = true;
y = false;
```

ω_4

Question: What is the probability of each of these worlds?

Your First Probabilistic Program

- Probability distribution for

```
x ~ flip(0.5);
y ~ flip(0.7);
```

```
x = true;
y = true;
```

ω_1

`0.5*0.7 = 0.35`

```
x = false;
y = true;
```

ω_2

`0.5*0.7 = 0.35`

```
x = false;
y = false;
```

ω_3

`0.5*0.3 = 0.15`

```
x = true;
y = false;
```

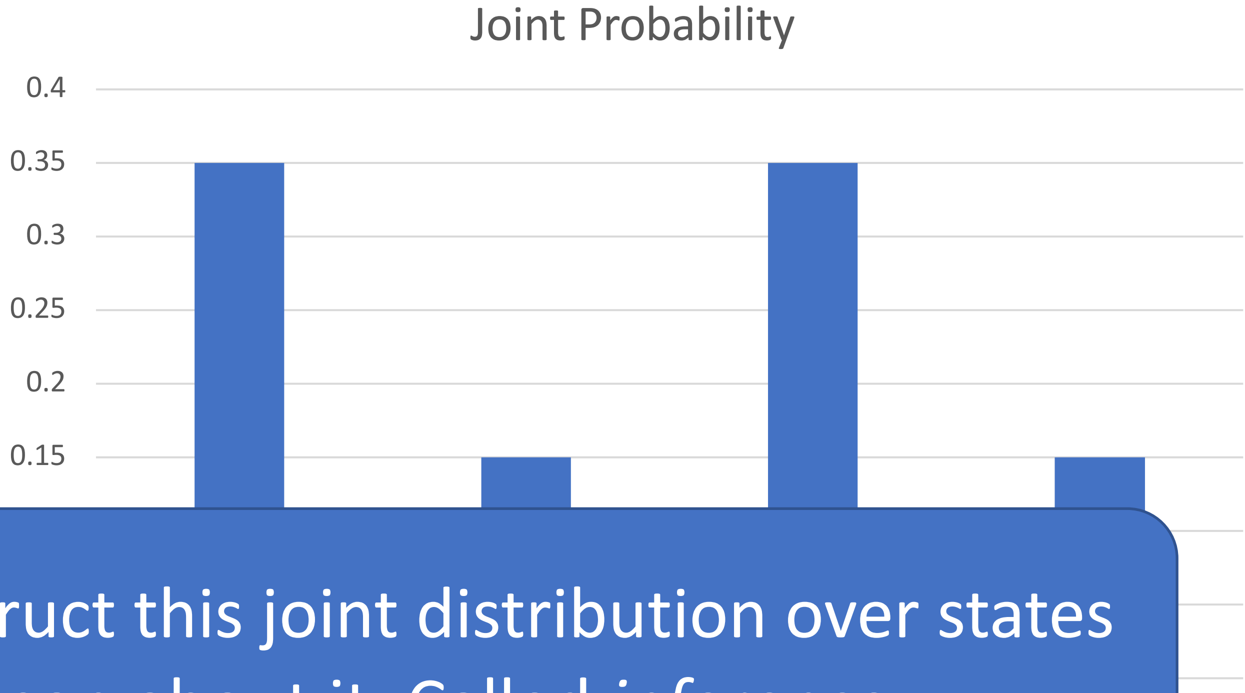
ω_4

`0.5*0.3 = 0.15`

Probabilistic Programs

- Use a *programming language to specify the distribution*

```
x ~ flip(0.5);
y ~ flip(0.7);
```

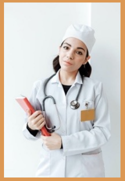


Goal: To construct this joint distribution over states and reason about it. Called *inference.*

PROGRAMMERS

Population: 100s of millions

People who use probabilistic models



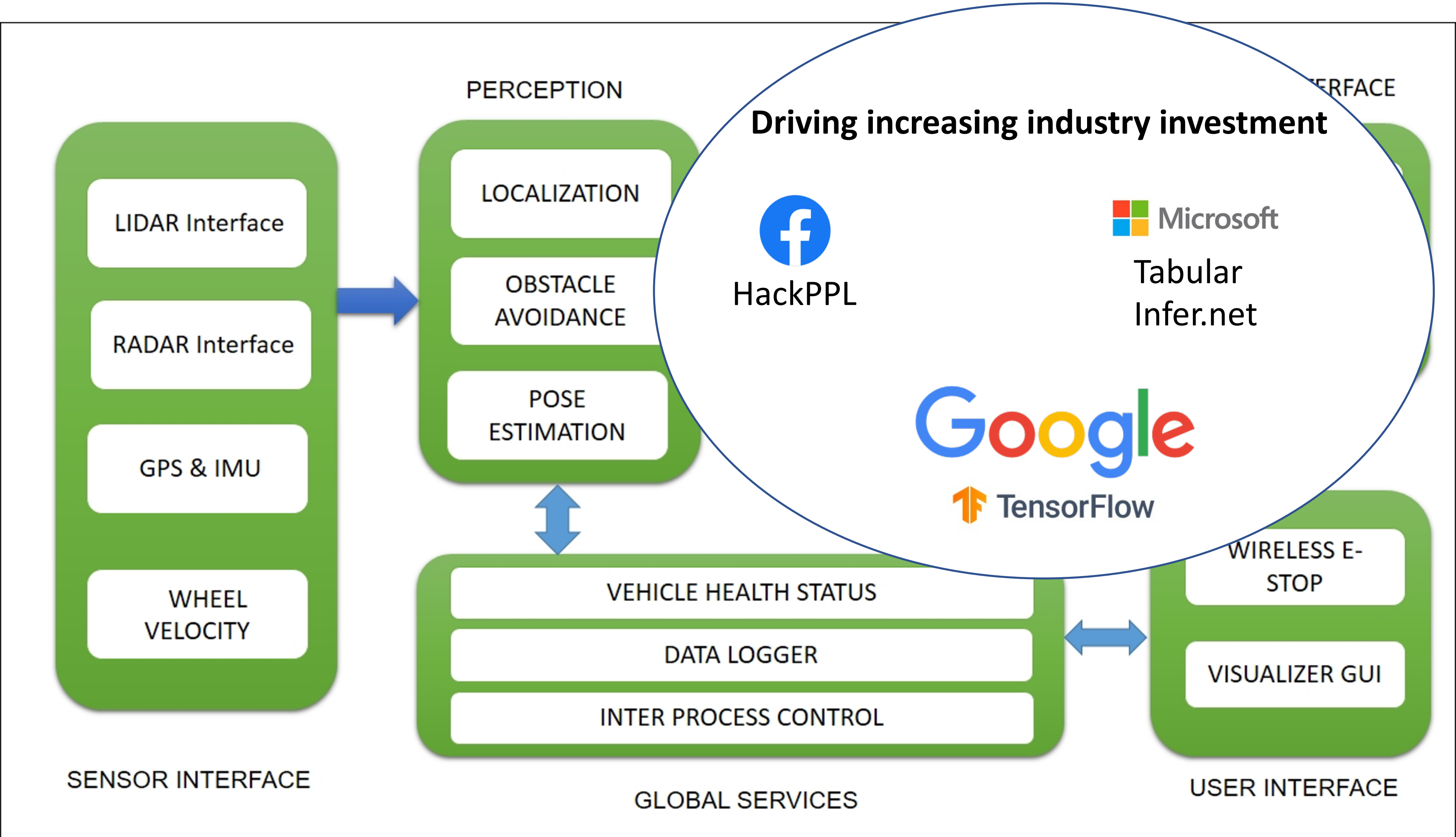
Google

Population: millions

Experts in probabilistic modeling

Population: thousands?

PPLs Enable Probabilistic Systems Design



Driving increasing industry investment

HackPPL

Microsoft

Tabular
Infer.net

Google

TensorFlow

PPLs are Popular

- Example: Stan, Tens of thousands of users across many domains (medicine, physics, astronomy, …)
- There are *many* more languages
- PROBPROG conference

Big Challenges in PPLs

1. Inference is *hard*
2. Language design is *hard*
3. Program analysis is *hard*
4. Programming (debugging, collaborating) is *hard*
5. *… we may find more throughout the course!*

Big Challenges in PPLs

- Solving these challenges will require insights from multiple fields…
  - AI/Statistics: Probabilistic inference and modeling, graphical models, Monte-Carlo methods, …
  - Programming languages: language design, compilers, program optimization, types and type systems, program analysis

- In this class we will touch on some of these ideas, build some of these foundations, and get a feel for what is left out there