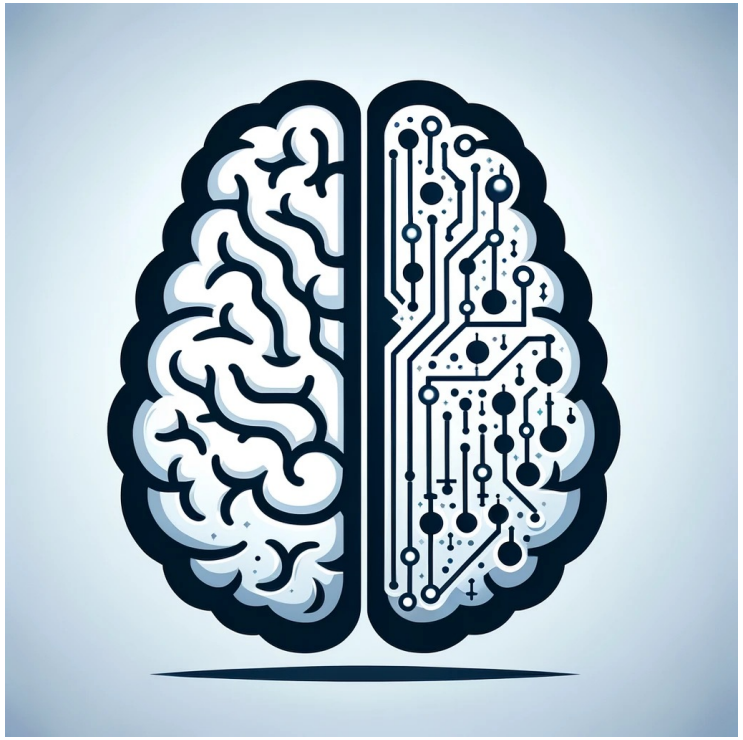
EN 601.473/601.673: Cognitive Artificial Intelligence (CogAI)



Lecture 6: PLOT, Bayesian Networks, probabilistic programs

Tianmin Shu

Recap: Language of thought

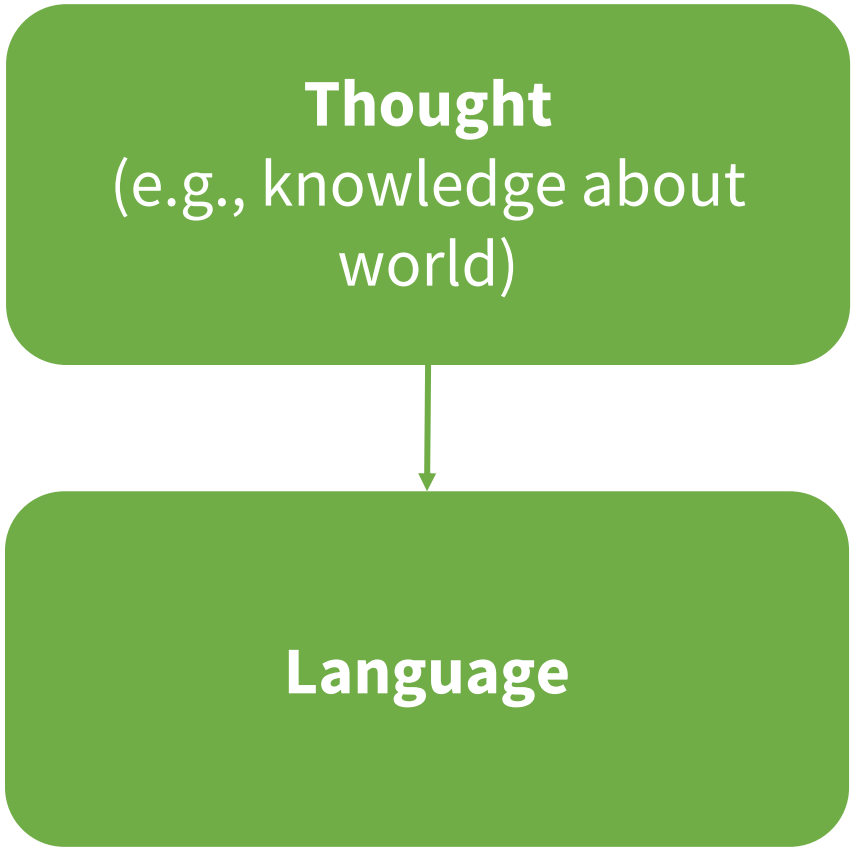
- Flexible & compositional hypothesis proposals
- Programs (e.g., lambda functions)
- Natural language

What can LLMs reason about in the number game?

- Propose a single hypothesis (typically a math property), and describe them in language

- Generate new numbers based on the hypothesis, with some level of uncertainty estimation

Why language models can reason?



Text data used for training language models

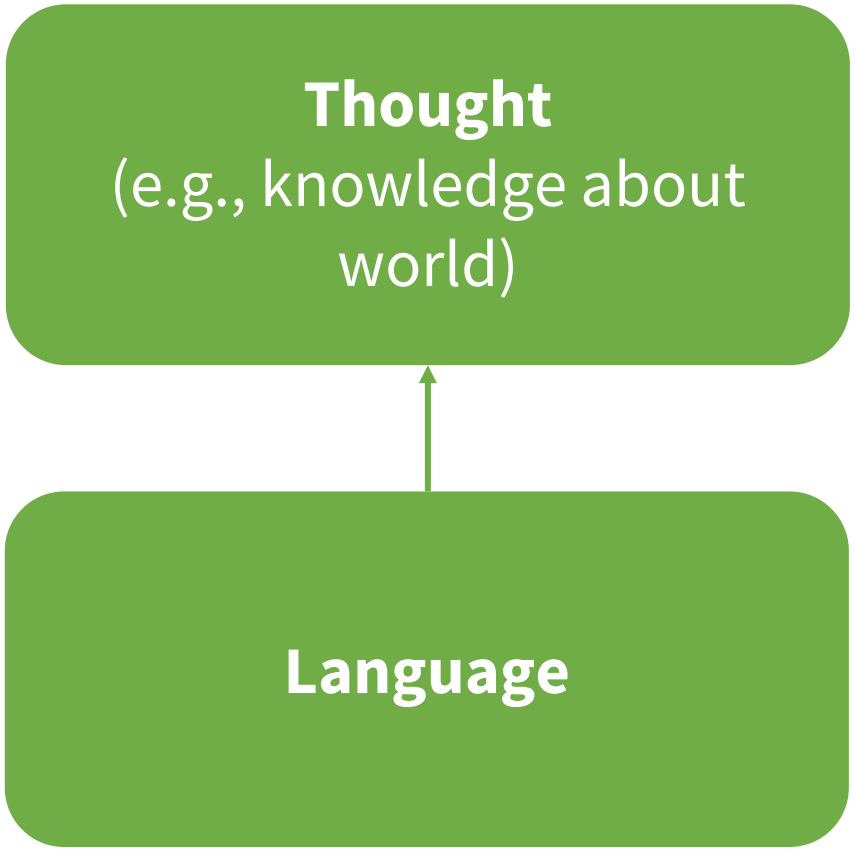
There are limits

- Not considering all possible hypotheses

- The confidence estimation could be brittle

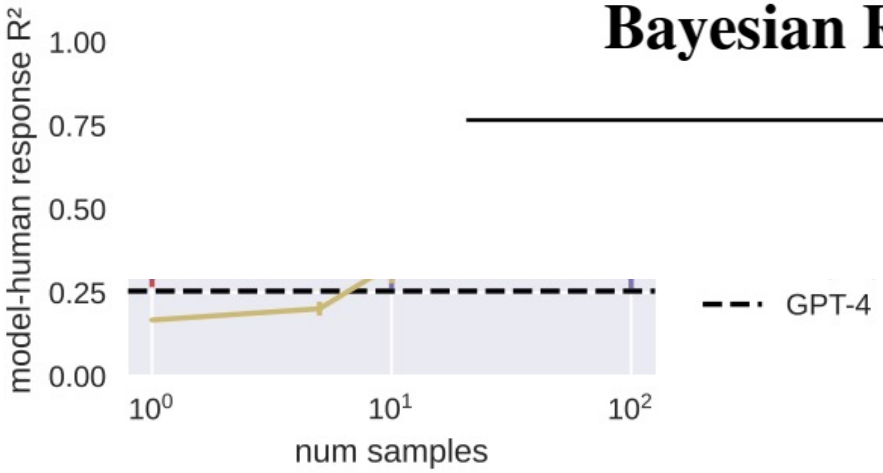
What about the reverse process?

- Natural language → thought (which guides Bayesian inference)
- Potentially combines the flexibility of language with the robustness of Bayesian inference



1. Programs
2. Priors

…



Human-like Few-Shot Learning via Bayesian Reasoning over Natural Language

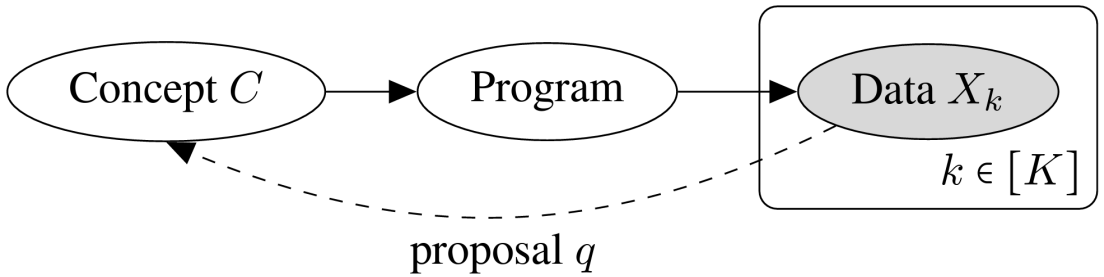
Kevin Ellis
Cornell University
kellis@cornell.edu

NeurIPS 2023

What's new compared to NeurIPS 1999

Abstract

A core tension in models of concept learning is that the model must carefully balance the tractability of inference against the expressivity of the hypothesis class. Humans, however, can efficiently learn a broad range of concepts. We introduce a model of inductive learning that seeks to be human-like in that sense. It implements a Bayesian reasoning process where a language model first proposes candidate hypotheses expressed in natural language, which are then re-weighed by a prior and a likelihood. By estimating the prior from human data, we can predict human judgments on learning problems involving numbers and sets, spanning concepts that are generative, discriminative, propositional, and higher-order.



Prior distribution. We consider two different prior distributions. The **pretrained prior** scores the log likelihood of each concept C using an open source language model (specifically, CodeGen 350M [41]). The **tuned prior** first extracts semantic features of the natural language concept C using a pretrained sentence feature extractor ϕ, specifically all-MiniLM-L6 [42], which outputs a 384-dimensional feature vector. The tuned prior maps those features to an (unnormalized) log probability via a linear mapping with parameters θ:

$$\text{Tuned prior:} \quad p_\theta(C) \propto \exp\left(\theta \cdot \phi\left(C\right)\right) \tag{6}$$

Likelihood distribution. Evaluating $p(X|C)$ requires first determining which numbers belong to the concept C. To efficiently enumerate those numbers, we translate C from natural language to python using Codex code-davinci-002 [22], a large language model trained on source code. We run the python code on the numbers 1..100 to determine the members of C. Given the members of C, we assume numbers are drawn uniformly at random from C with probability $(1-\epsilon)$, and uniformly at random from 1..100 with probability ϵ:

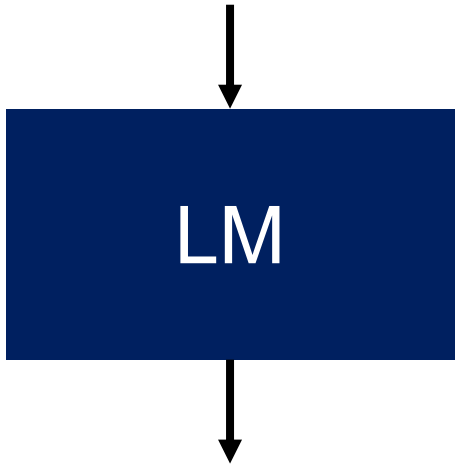
$$p(X|C) = (1-\epsilon)\frac{\mathbb{1}\left[X \in C\right]}{|C|} + \epsilon\frac{1}{100} \tag{7}$$

Proposal distribution. We implement q using Codex code-davinci-002 [22]. We prompt Codex by adapting the cover story given to the human subjects, then append the training example numbers $X_{1:K}$ and have it complete the natural language description of the hidden concept. We used Codex because we hypothesized that training on source code would transfer to reasoning about numbers.

Temperature, Platt transform. Because human subjects rated on a scale of 1-7, we introduce a learnable Platt transform between the model's predicted probabilities and the human judgments [43]. We also place a learnable temperature parameter on the posterior.

Sampling natural language hypothesis proposals from LMs

Prompt: cover story, example numbers



between 30 and 45

even numbers

…

Pretrained prior:
Log likelihood of the concept produced by an LM

Tuned prior:
Learn a prior based on language features

$$p_\theta(C) \propto \exp\left(\theta \cdot \phi\left(C\right)\right)$$

Use some of the human data as training data
Test on unseen human data

Natural language hypothesis to program

- LLMs are good at writing code

You

Write python code that generates all integers from 30 to 45.

GPT-4

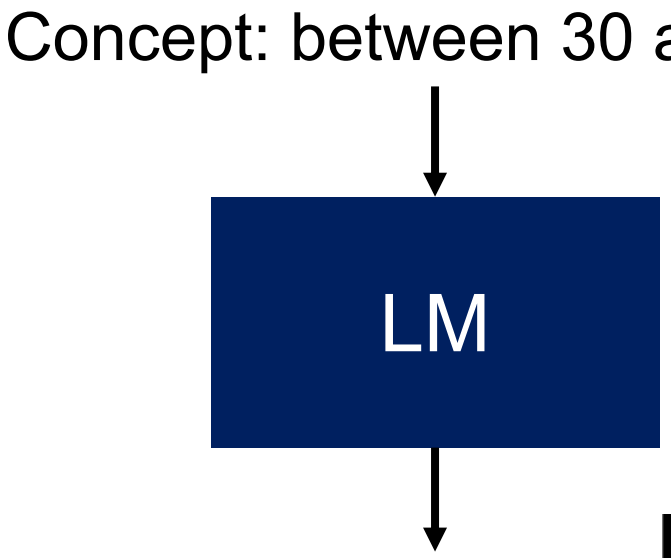
python Copy code

```
# Generating all integers from 30 to 45
integers = list(range(30, 46))
integers
```

Result

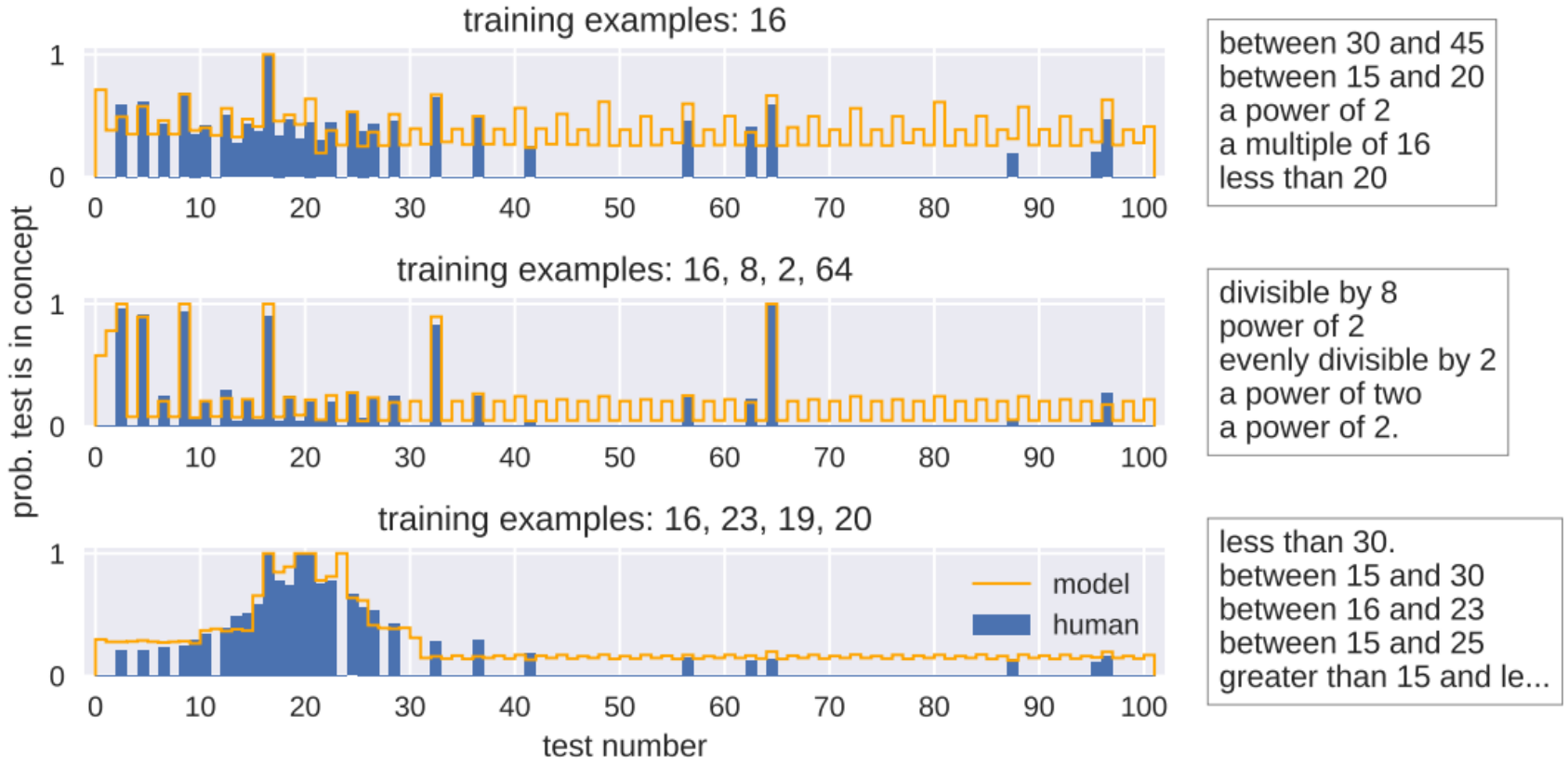
```
[30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
```

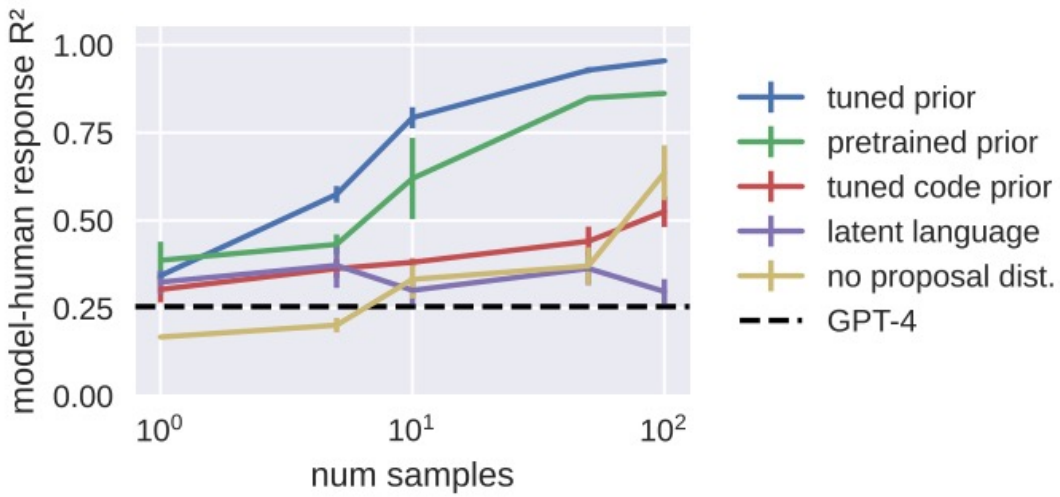
Natural language hypothesis to program

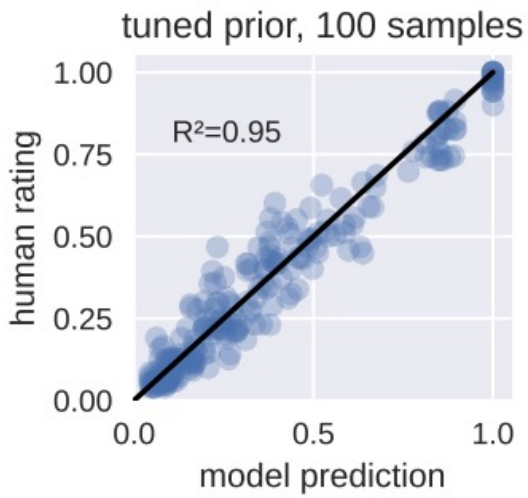



```
# Generating all integers from 30 to 45
integers = list(range(30, 46))
integers
```

Run the code on x = 1..100 to generate all numbers that fit the concept
→ Likelihood based on the size principle

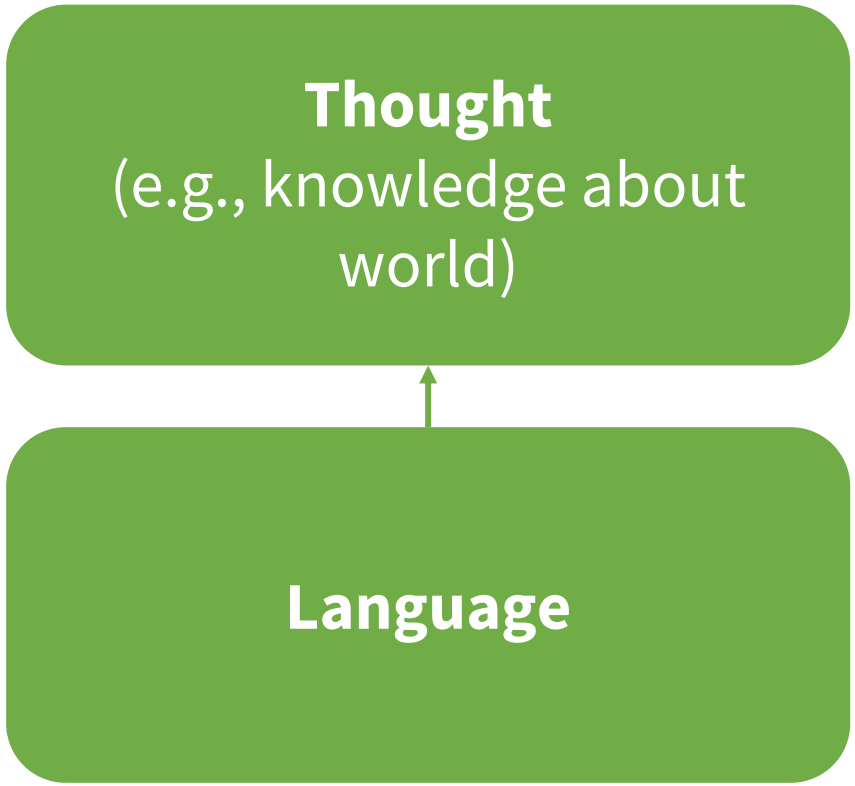

training examples: 16
between 30 and 45
between 15 and 20
a power of 2
a multiple of 16
less than 20
training examples: 16, 8, 2, 64
divisible by 8
power of 2
evenly divisible by 2
a power of two
a power of 2.
training examples: 16, 23, 19, 20
less than 30.
between 15 and 30
between 16 and 23
between 15 and 25
greater than 15 and le...
prob. test is in concept
test number
model
human


model-human response R²
num samples
tuned prior
pretrained prior
tuned code prior
latent language
no proposal dist.
GPT-4


tuned prior, 100 samples
R²=0.95
human rating
model prediction

Possible extensions

- Language of thought → probabilistic language of thought
- Composing more complex concepts (physical and social concepts)
- A more general approach to calibrate an LLM-based model's uncertainty with human uncertainty



Taking stock (Bayesian concept learning)

- Principled cognitive models → better AI models
- Experiments used to evaluate human intelligence → AI benchmarks for machine intelligence
- Stronger AI models (neuro-symbolic models) can potentially give us new accounts for human intelligence
  - What priors / inductive biases could be learned & how
  - Why people can conduct both robust and rapid judgment?
    - Robustness: symbolic & Bayesian reasoning
    - Speed: neural networks

Towards a probabilistic language of thought

- Bayesian networks

- Probabilistic programs

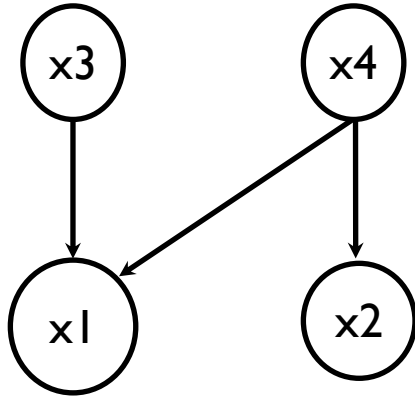
Languages for probabilistic models

- (Languages not just for the hypotheses but for the whole models)
- Why a language?
  - Makes writing down models easier
  - Makes reasoning about models clearer and more flexible
  - Gives idea about the mind's representation and algorithms (probabilistic language of thought)
  - Utilize the openendedness and flexibility of large language models (LLMs)

- Languages we will discuss
  - Directed graphical models (e.g., Bayesian nets)
  - Undirected graphical models (e.g., energy-based models or EBMs)
  - Probabilistic programs
    - Probabilistic programming languages (PPL): Church, Venture, Webppl, Gen, Edward, Stan, PyMC, Figaro, Anglican, BLOG, Pyro, ProbTorch, TensorFlow Probability (BayesFlow), MetaGen, GenJAX…
    - Webppl: probmods.org, an educational tool
    - Gen: Gen.dev, a state-of-the-art PPL

Bayesian networks

- A Bayesian network (or Bayes net) is a way to specify a joint distribution: given the (in)dependence structure, then the conditional distributions
- Formal definition: A *Bayesian net*work:
  - A set of *n* variables V = {x_1, …, x_n}
  - A directed acyclic graph (DAG) on V.
  - A local conditional distribution for each variable in V given its parents in the graph (a factor in the joint dist.).



$$P(x_1,\ldots,x_n) = \prod_i P(x_i \mid Parents[x_i])$$

- A graph with n nodes and m edges, how many factors?
- Factorization simpler than the full joint distribution, which yields efficient learning and inference.

Example

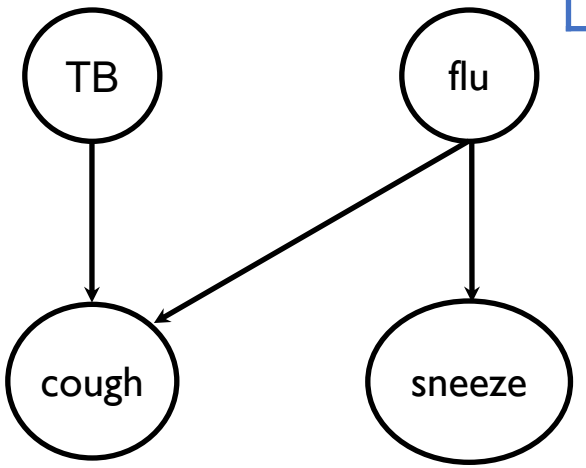
- Medical diagnosis model:
  - Symptoms: cough, sneeze
  - Diseases: tuberculosis (TB) , flu
  - 4 binary variables
  - P(cough, sneeze, TB, flu)

P(TB)

TB = t
0.1

P(flu)

flu = t
0.2



Conditional probability table (CPT)

P(cough | TB, flu)

TB	flu	cough = t
t	t	0.9
t	f	0.8
f	t	0.75
f	f	0.1

P(sneeze | flu)

flu	sneeze = t
t	0.8
f	0.2

P(cough, sneeze, TB, flu)
=P(cough | TB, flu) x P(sneeze | flu) x P(TB) x P(flu)

Model specified by 8 numbers + graph structure vs. 15 for the full joint distribution …

Example

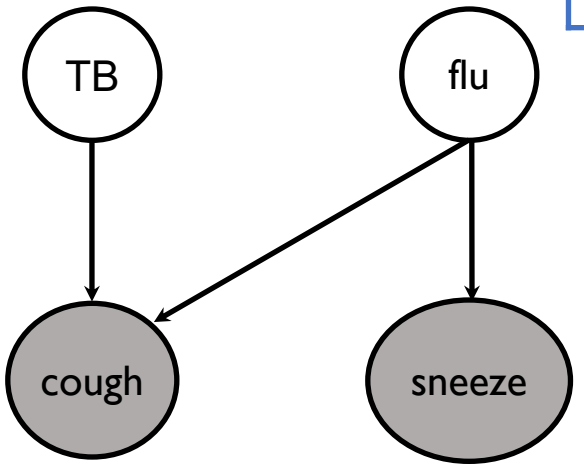
- Medical diagnosis model:
  - Symptoms: cough, sneeze
  - Diseases: tuberculosis (TB) , flu
  - 4 binary variables
  - P(cough, sneeze, TB, flu)

P(TB)

TB = t
0.1

P(flu)

flu = t
0.2



P(cough | TB, flu)

TB	flu	cough = t
t	t	0.9
t	f	0.8
f	t	0.75
f	f	0.1

P(sneeze | flu)

flu	sneeze = t
t	0.8
f	0.2

P(cough, sneeze, TB, flu)
=P(cough | TB, flu) x P(sneeze | flu) x P(TB) x P(flu)

Inference: compute conditional probabilities of ***latent*** variables given ***observable*** variables, e.g., P(flu | cough, sneeze)

Learning: infer parameters or structure from data

Example

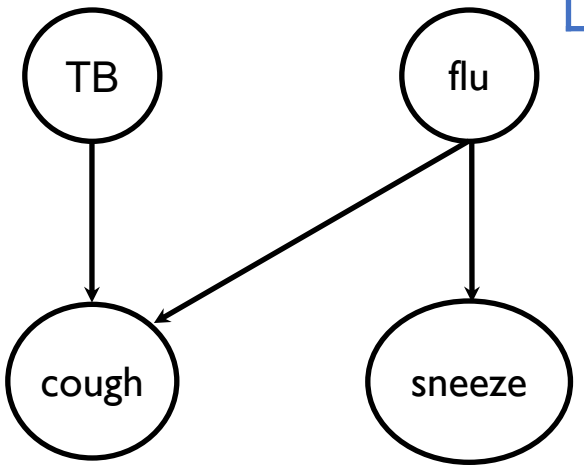
- Medical diagnosis model:
  - Symptoms: cough, sneeze
  - Diseases: tuberculosis (TB) , flu
  - 4 binary variables
  - P(cough, sneeze, TB, flu)

P(TB)

TB = t
0.1

P(flu)

flu = t
0.2



P(cough | TB, flu)

TB	flu	cough = t
t	t	0.9
t	f	0.8
f	t	0.75
f	f	0.1

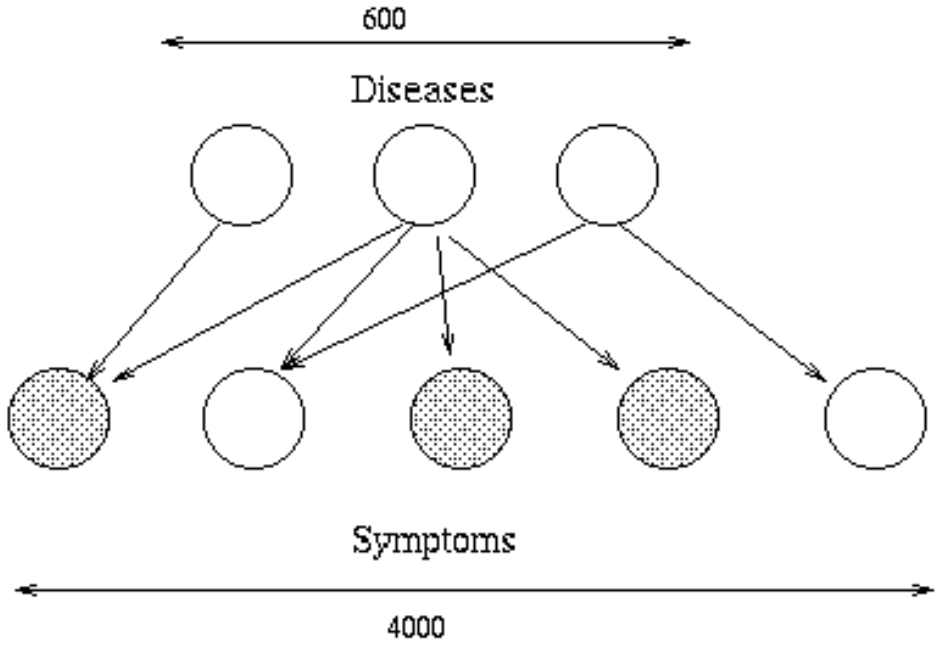
P(sneeze | flu)

flu	sneeze = t
t	0.8
f	0.2

Only a modest gain in this simple example

P(cough, sneeze, TB, flu)
=P(cough | TB, flu) x P(sneeze | flu) x P(TB) x P(flu)

Quick medical reference (QMR) model



Now joint distribution in many fewer numbers!

- P(*m*) ~600 (if priors of diseases are independent)
- P(*s*|*m*) ~4000*2^k (if a symptom might be caused by k diseases, on average)
- Compare with 2^{4600}!

Special properties that simplify probability computation

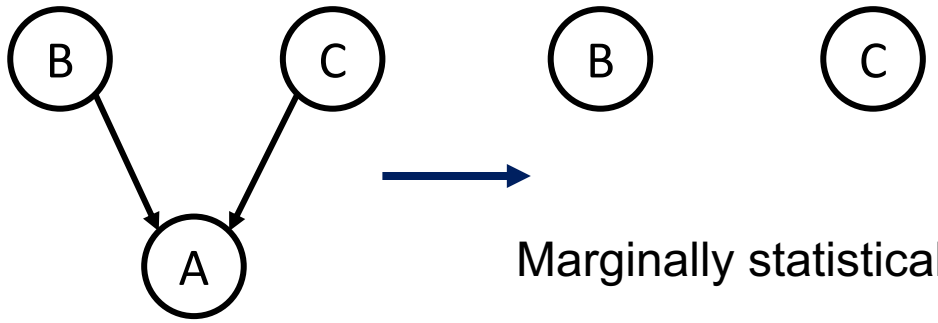
- A key property: locally normalized
  - Each local conditional distribution satisfies:

$$\sum_{x_i} P(x_i | Parents[x_i]) = 1$$

- Implications:
  - Consistency of sub-Bayesian networks
  - Consistency of local conditionals

Consistency of sub-Bayesian networks

Marginalization of a leaf node yields a Bayesian network without the node



Marginally statistically independent

$$P(B,C) = \sum_A P(A,B,C)$$

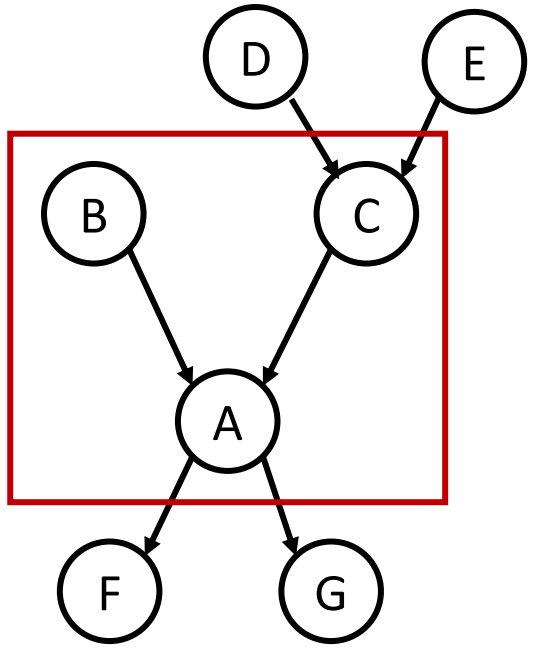
$$= \sum_A P(B)P(C)P(A|B,C)$$

$$= P(B)P(C)\sum_A P(A|B,C)$$

$$= P(B)P(C)$$

Consistency of local conditionals

Local conditional distributions are the true conditional distributions (no need to consider nodes other than the parent nodes)

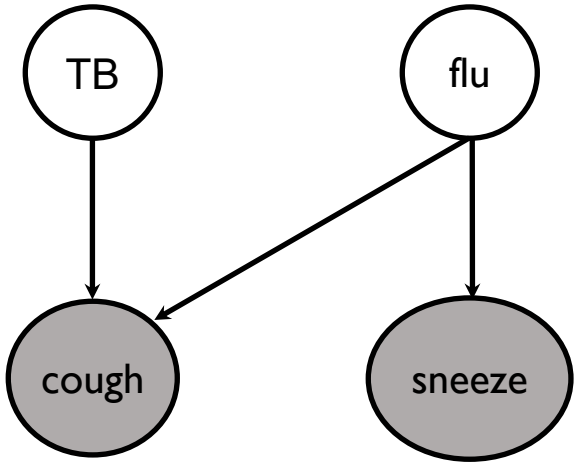


$$P(A \mid B, C) = \sum_{all\ other\ nodes} P(A, all\ other\ nodes \mid B, C)$$

$$= P(A \mid B, C)$$

Probabilistic inference with Bayesian networks

- Given:
  - Bayesian network that defines a joint distribution
    $$P(x_1, x_2, \ldots, x_n)$$
  - Condition / evidence / observation
    $O = o, O \subseteq X$ is a subset of (observed) variables
  - Query:
    $Q \subseteq X$ is a subset of (latent) variables
- Compute:



$P(Q = q | O = o)$ for all values q

Inference algorithms

- Exact inference for simple networks
  - Variable elimination algorithm, dynamic programming, message passing…
- Sampling-based inference
  - Approximate inference, but can work for any network, as well as more expressive probabilistic models
  - Importance sampling, MCMC, SMC
- Amortized / compiled inference (learned inference)
  - Can use neural networks or other data-driven ML methods

Why Bayesian networks

- Mathematically convenient
- Statistical (in)dependence introduced by a graph structure

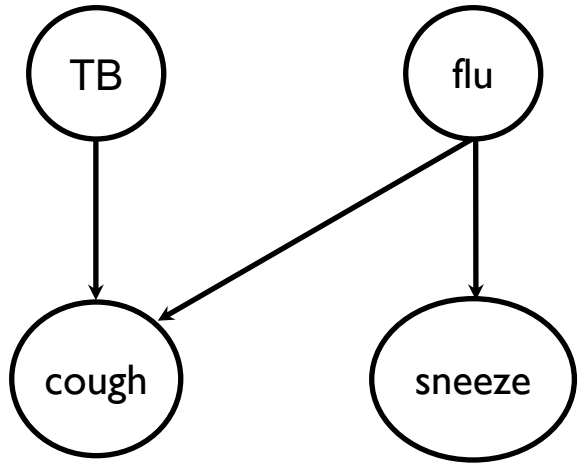
- Causal relations defined by a graph structure
- Bayesian networks describe causal processes
- Model causal reasoning via Bayesian inference

Judea Pearl

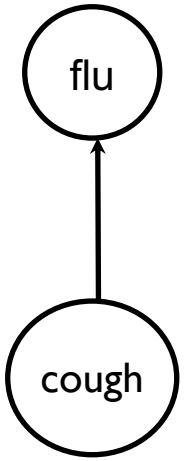
2011 Turing Award: For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning.

Causality

- A Bayesian network describe a causal process,
  or a causal model of the world



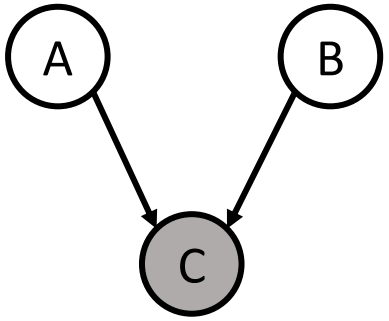
- The sequence of choices that must be made to construct a world
- This is not the flow of time, but of dependence
- Important for expressing knowledge of how the world works
- Note: statistical dependence does not mean causal dependence
  - Mathematically correct doesn't mean causally correct
- Causal Inference (601.477/677)
- In this course, we will discuss how commonsense knowledge of the world and other agents can be expressed by (causal) Bayesian networks
- Model-based reasoning → causal reasoning → Bayesian inference



Bayesian networks capture causal reasoning patterns

A	B	C = t
t	t	0.9
t	f	0.9
f	t	0.9
f	f	0.2

Explaining away

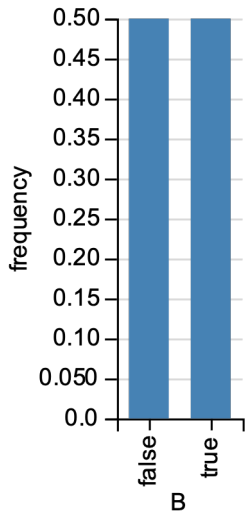


A=t
0.5

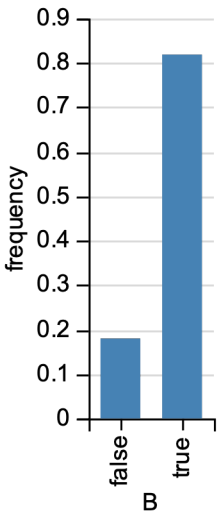
B=t
0.5

$$P(B|A,C) = \frac{P(C|A,B)P(B)}{\sum_B P(C|A,B)P(B)}$$

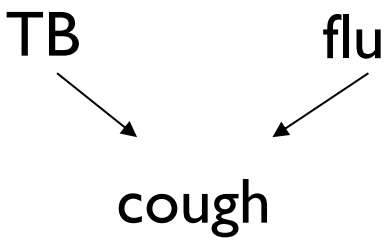
$P(B|A=t,C=t)$

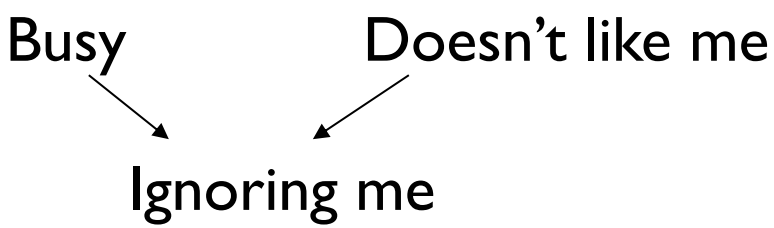


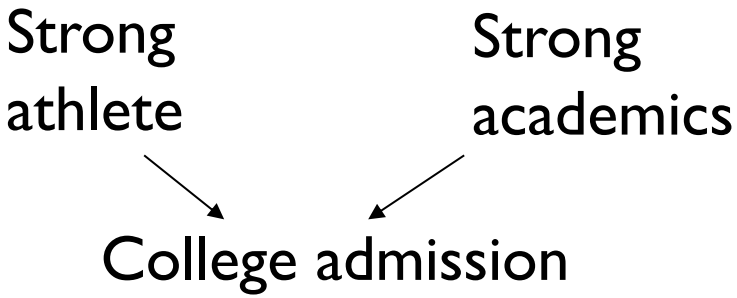
$P(B|A=f,C=t)$

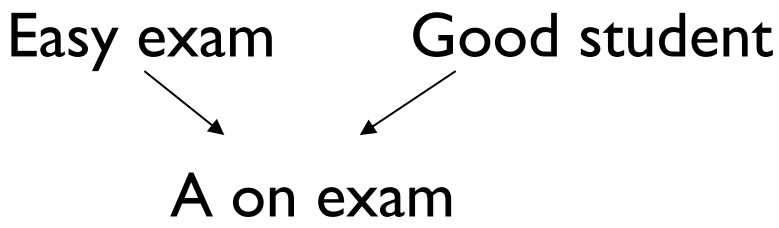


Explaining away in cognition

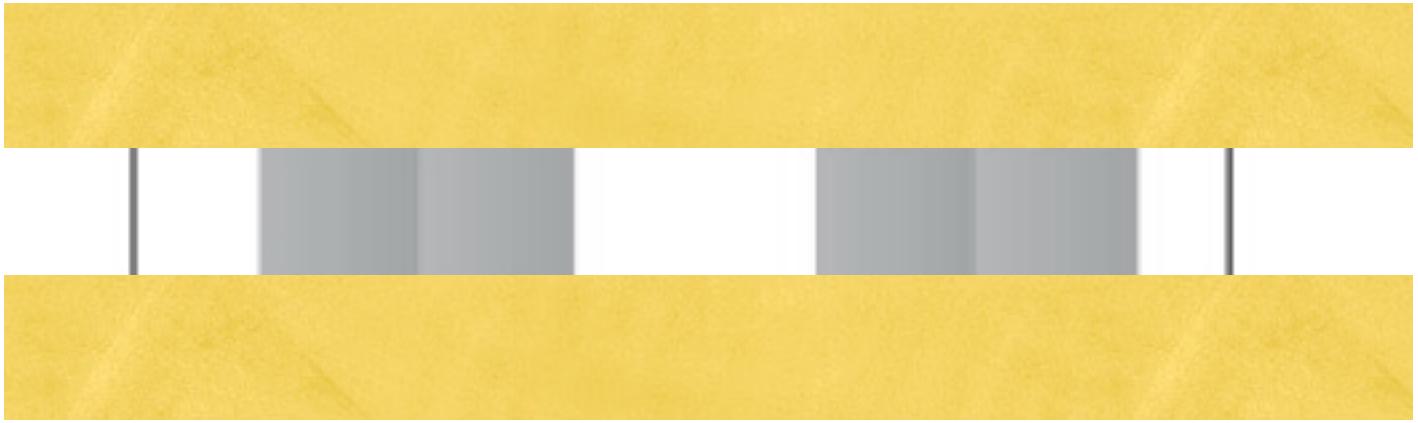








Explaining away in vision



Explaining away in vision

The Mach Illusion…

Casual model