Computational Cognitive Modeling

course website:
https://brendenlake.github.io/CCM-site/

Computational Cognitive Modeling

course website:
https://brendenlake.github.io/CCM-site/

Brenden Lake

Assistant Professor, Data Science and Psychology

office hours: Mondays 4:30-5:30pm, 60 5th Ave. (CDS) Room 610

https://cims.nyu.edu/~brenden
https://lake-lab.github.io/

Todd Gureckis

Professor, Psychology

Affiliate, Center for Data Science

office hours: TBD

http://gureckislab.org

Solim LeGris

PhD student, Psychology

office hours: TBD

Cindy Luo

MS student, Data Science

office hours: TBD

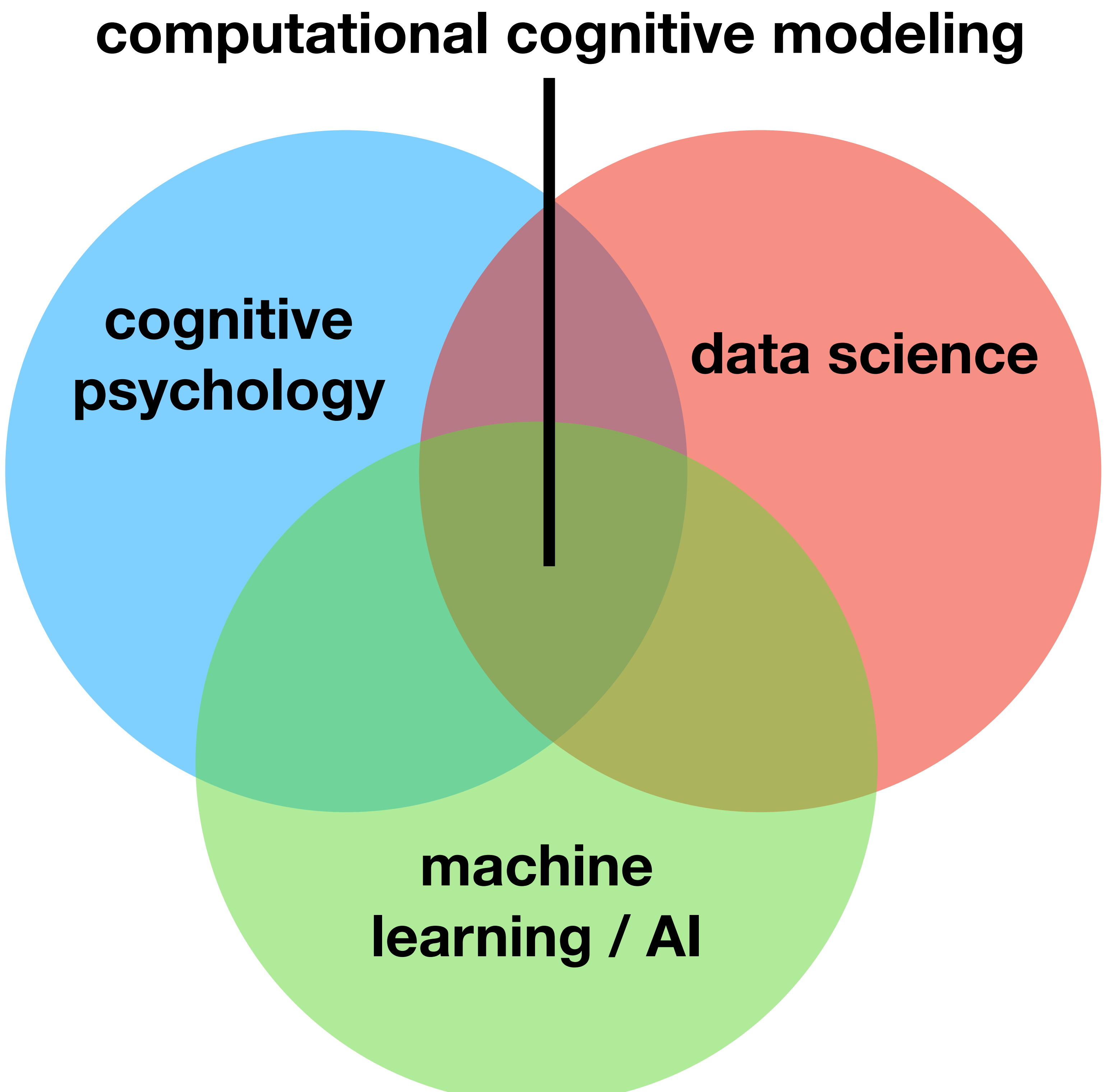
What is Computational Cognitive Modeling?

- **Computational Cognitive Modeling** is devoted to understanding the human mind and brain, in terms of their underlying computational processes.
- Building computer simulations that *mimic* the intelligent behavior of humans, and using these simulations to predict and explain human behavior.

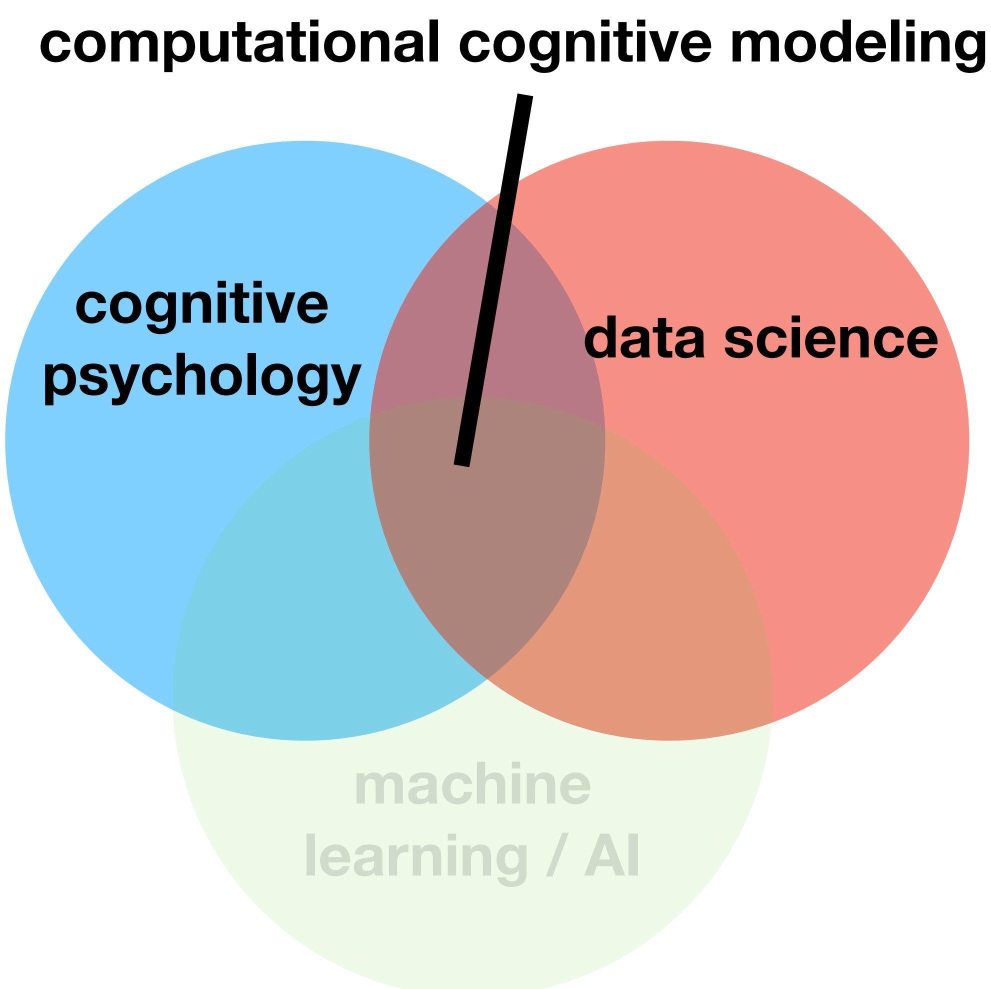
Key questions for this course

- What is intelligence?
- What kind of computer is the mind and brain?
- Can we better understand the mind/brain by building computational cognitive models?
- Can we better understand behavioral data by building computational cognitive models?
- Can we improve machine intelligence by incorporating insights from human intelligence?

At the intersection of cognitive psychology and data science

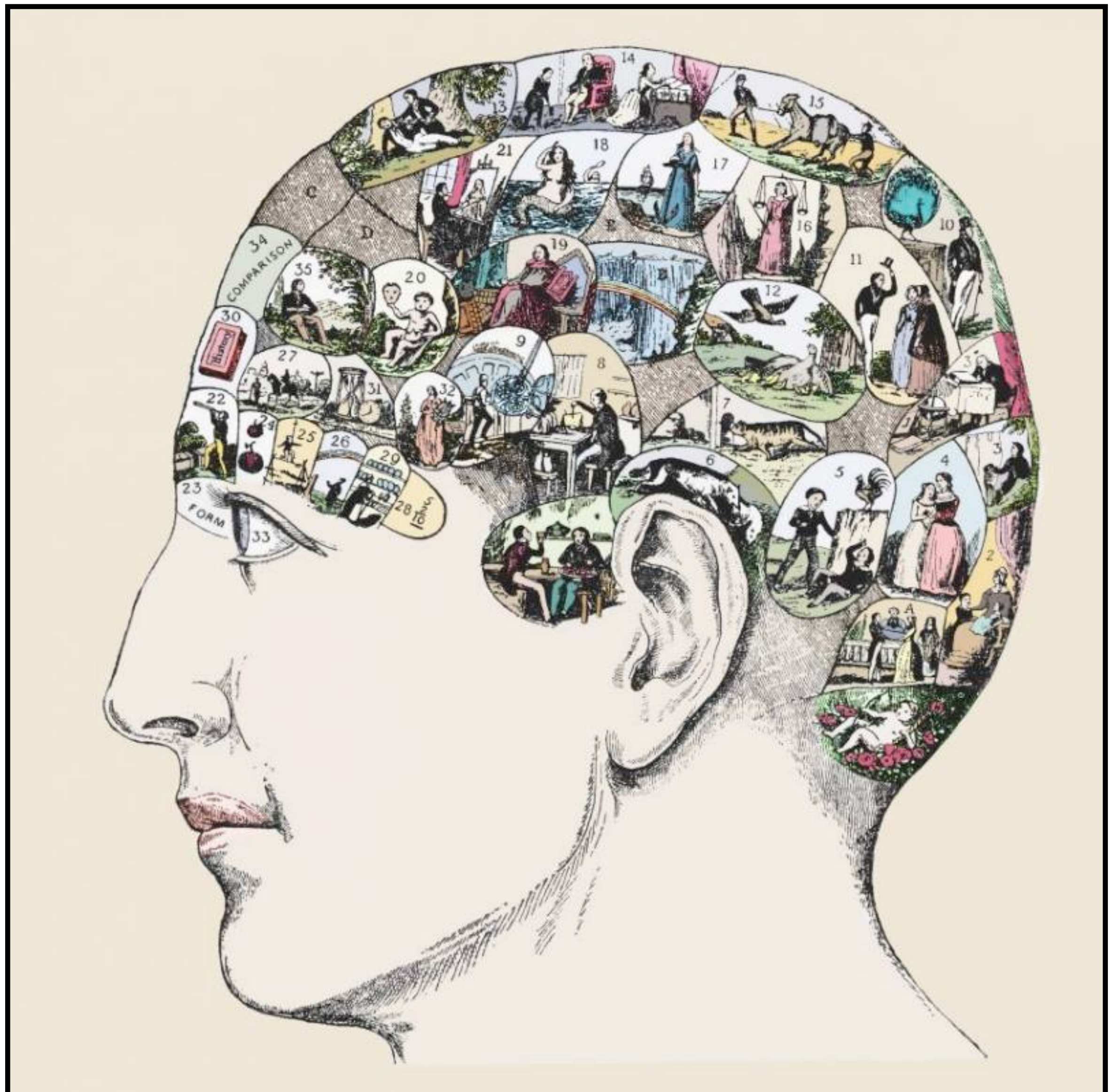


Connections between computational cognitive modeling and data science



- **Similar goals**: build computational models to explain or predict behavioral data
- **Similar computational paradigms and techniques**: neural networks / deep learning, reinforcement learning, Bayesian modeling, probabilistic graphical models, program induction
- Data science is about **extracting knowledge from data**. The human mind is the best (known) system for extracting knowledge from data.
- There is ripe potential for even deeper connections. We hope that, by bringing together students from a variety of backgrounds, this class can help realize this potential.

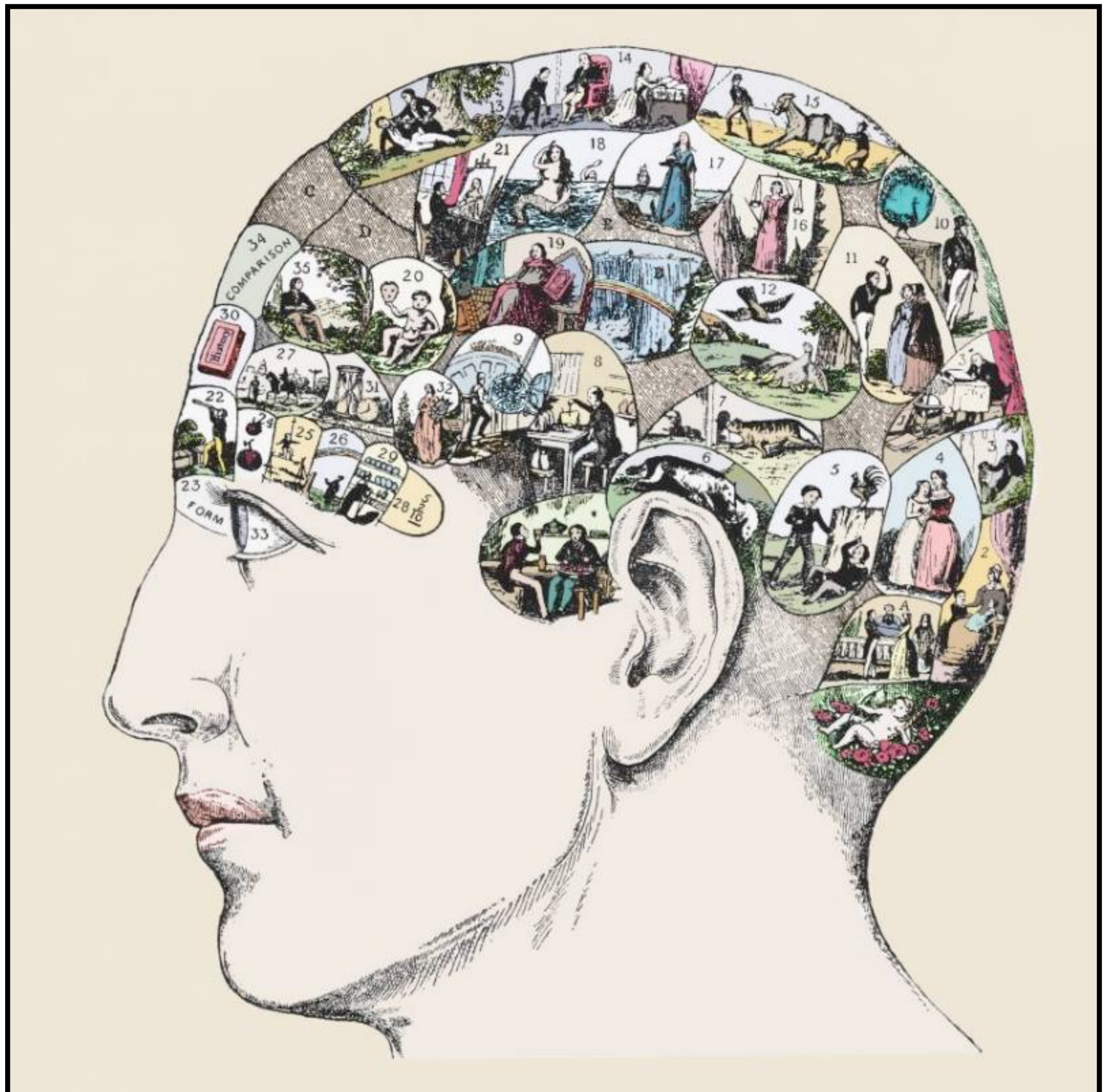
What is a mind?



SYMBOLICAL HEAD.

This has been debated for thousands of years. If you don't have an immediate answer, don't feel bad. Various proposals have been thrown around from by Plato, Buddha, Aristotle, Zoroaster…. ancient Greek, Indian, and Islamic philosophers, and even several folks at NYU.

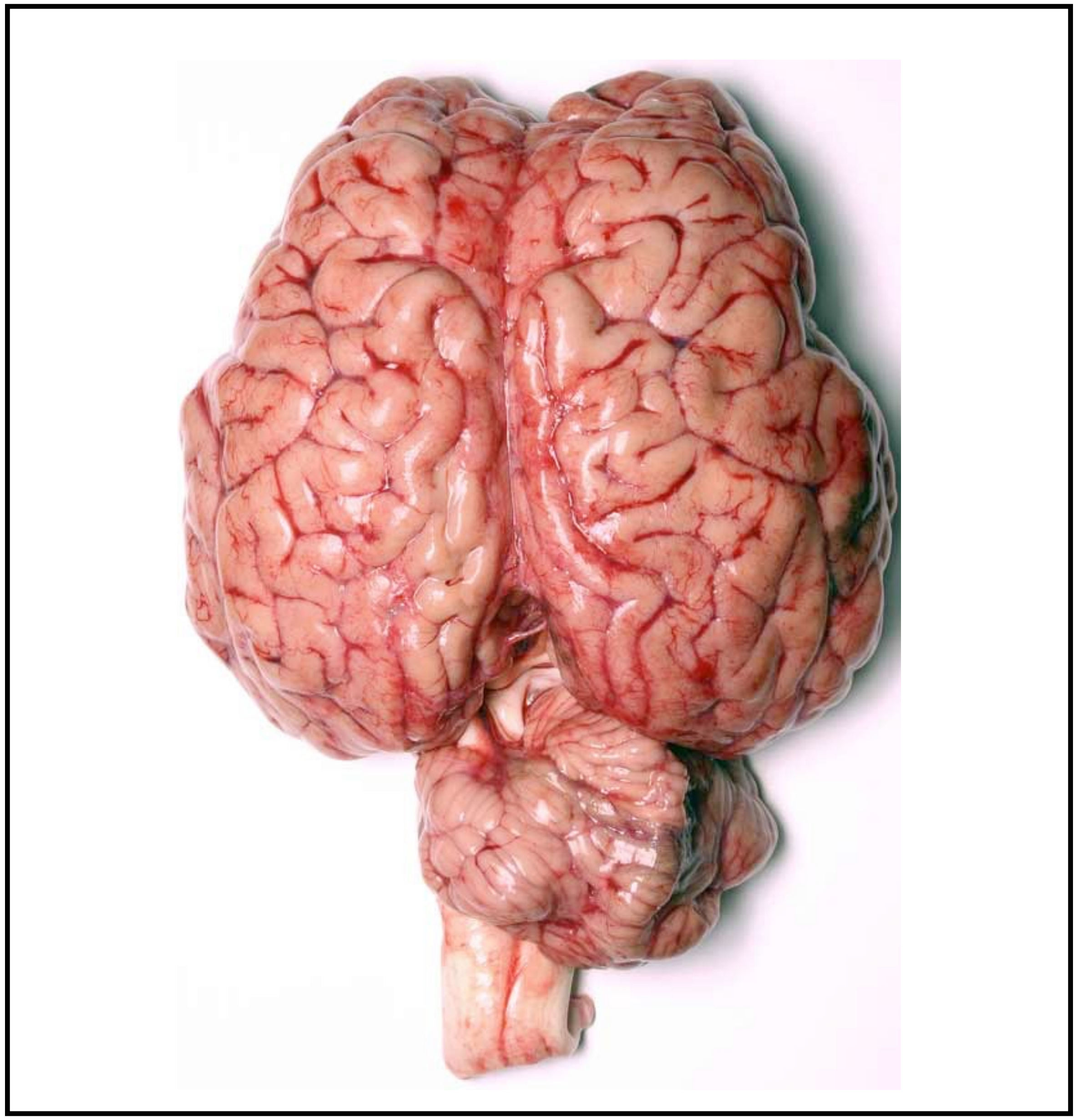
What is a mind?


SYMBOLICAL HEAD.

What do minds do?

Minds encompass our thoughts, which are mental processes that allow us to deal with the world. These include not only explicit wishes, desires, and intentions, but also unconscious processes.

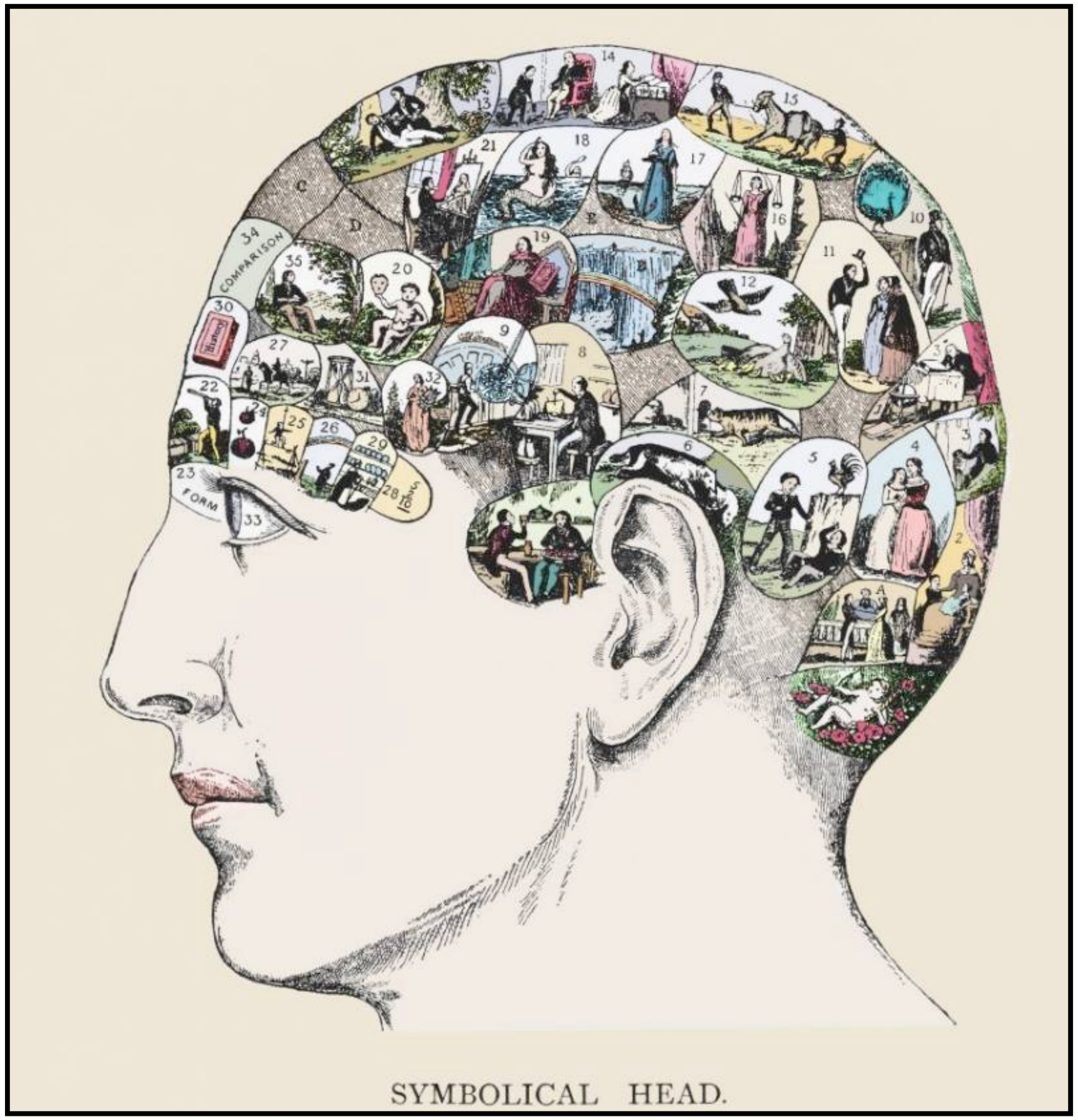
What is a mind?



Does MIND=BRAIN?

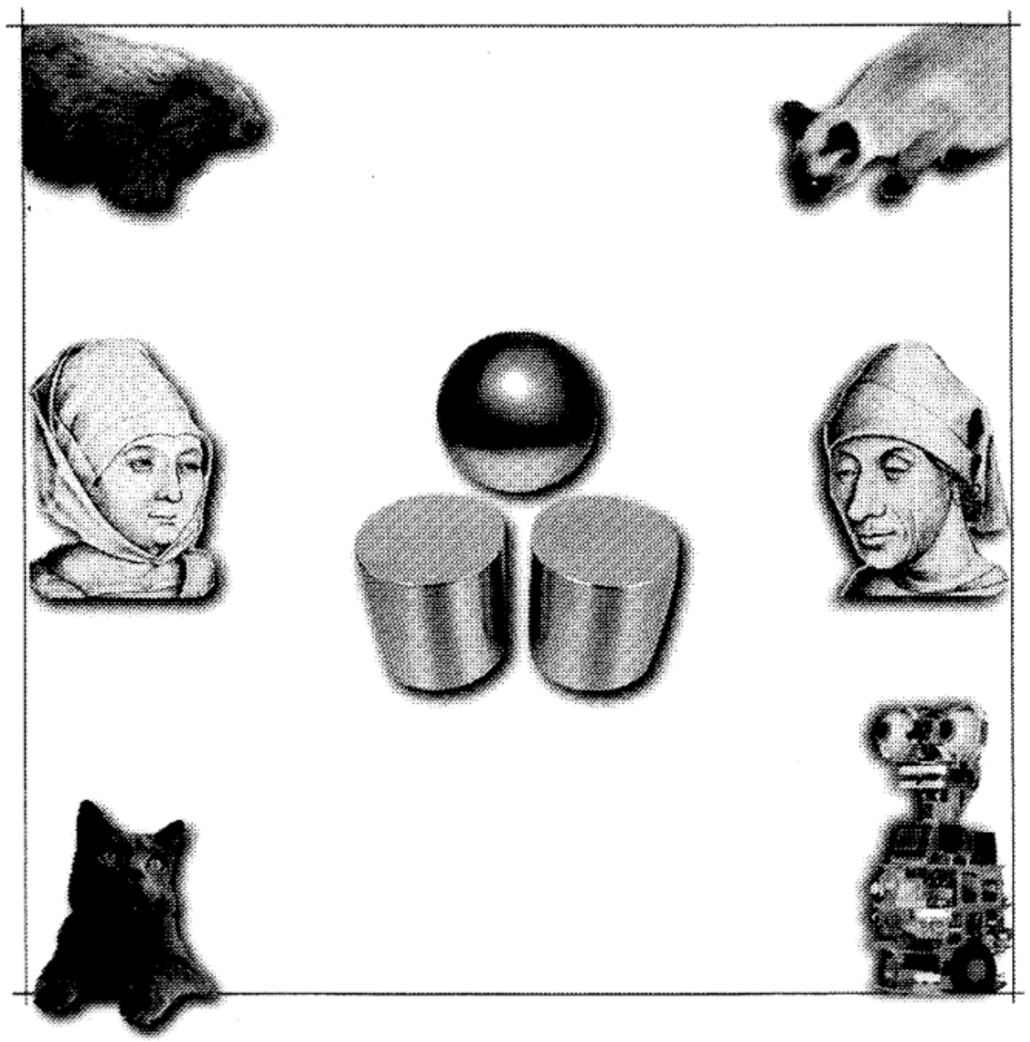
We know that we can’t have a mind or thoughts without a brain, but does that mean that minds and brain are synonymous?

What is a mind?



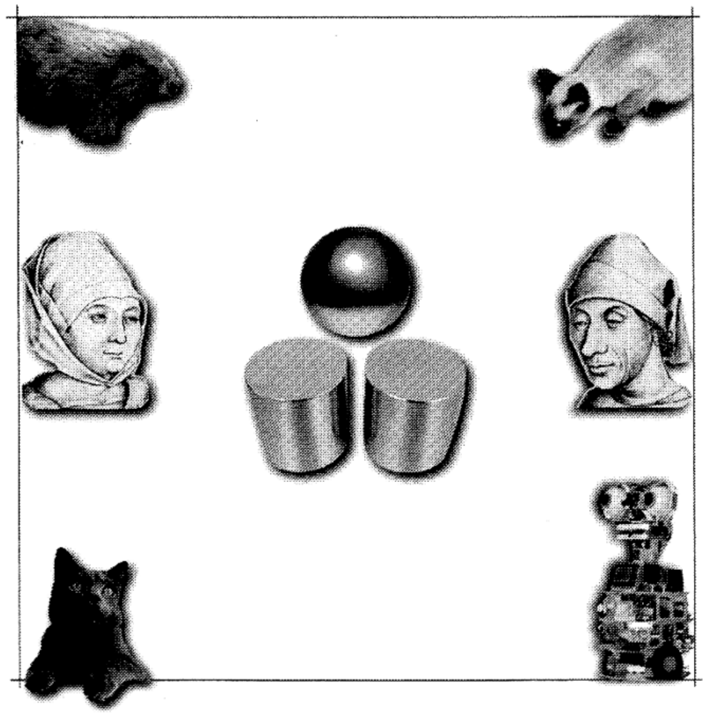
A "slippery slope" argument can convince us that minds are not literally brains, but encompass anything that is organized as representational states that accurately reflect aspects of the world.

The Brain/Mind Riddle



What is common to the various entities (person 1, person 2, cat 1, cat 2, robot, etc.) that look at this scene of two cylinders and a sphere and agree upon what is viewed?

Shimon Edelman's argument



The question: What is common to observers viewing the same scene and who agree upon what is viewed?

- Can't literally be neurons. My neurons are my own, and you can't borrow them to solve your own problems.

- Is it the literal organization of the human nervous system? We know (or at least believe) that cats have a very similar visual system and view the world much like we do. Is it the mammalian visual system? What about other animals?

- What about artificial systems formed of computers and video cameras that can accurately recognize the scene as well?

- **The key to minds is not their physical substrate, but the relations that states of the system have to one another, and to the external environment.**

Minds as computers

- Minds aren't human neurons or cat neurons or robot parts. They are dynamic, continually evolving systems that relate ongoing internal (i.e., mind) states and external (i.e., world) states
- Correspondences can be made between two systems by describing what they do, independent of their exact physical substrate.
- **We can describe these correspondences through the language of computation, simply because the THEORY OF COMPUTATION offers formal insights into how ostensibly dissimilar systems can be formally identical.**

Why build computational cognitive models? (As a psychologist)

"Verbally expressed statements are sometimes flawed by internal inconsistencies, logical contradictions, theoretical weaknesses and gaps. A running computational model, on the other hand, can be considered as a sufficiency proof of the internal coherence and completeness of the ideas it is based upon." (Fum, Del Missier, & Stocco, 2007)

Some famous psychological theories…

- Attention is like a spotlight
- A child learning about the world is like a scientist theorizing about science
- Language influences thought
- Working memory is having 7 +/- 2 slots to store items
- Categorization happens by comparing novel instances to past exemplars
- Categories influence perception

Each of these theories benefits from formalization with a computational model to…

- **Make predictions explicit**
- Implications often **defy expectations**
- **Aid communication** between scientists
- Support **cumulative progress**

“Formal (i.e., mathematical or computational) theories have a number of advantages that psychologists often overlook. They force the theorist to be explicit, so that assumptions are publicly accessible and reliability of derivations can be confirmed...” (Hintzman, 1990)

Rich history of connections between fields

cognitive science / psychology

⟶

⟵

machine learning / AI / data science

Bi-directional exchanges of computational methods and paradigms

cognitive science / psychology ⇄ **machine learning / AI / data science**

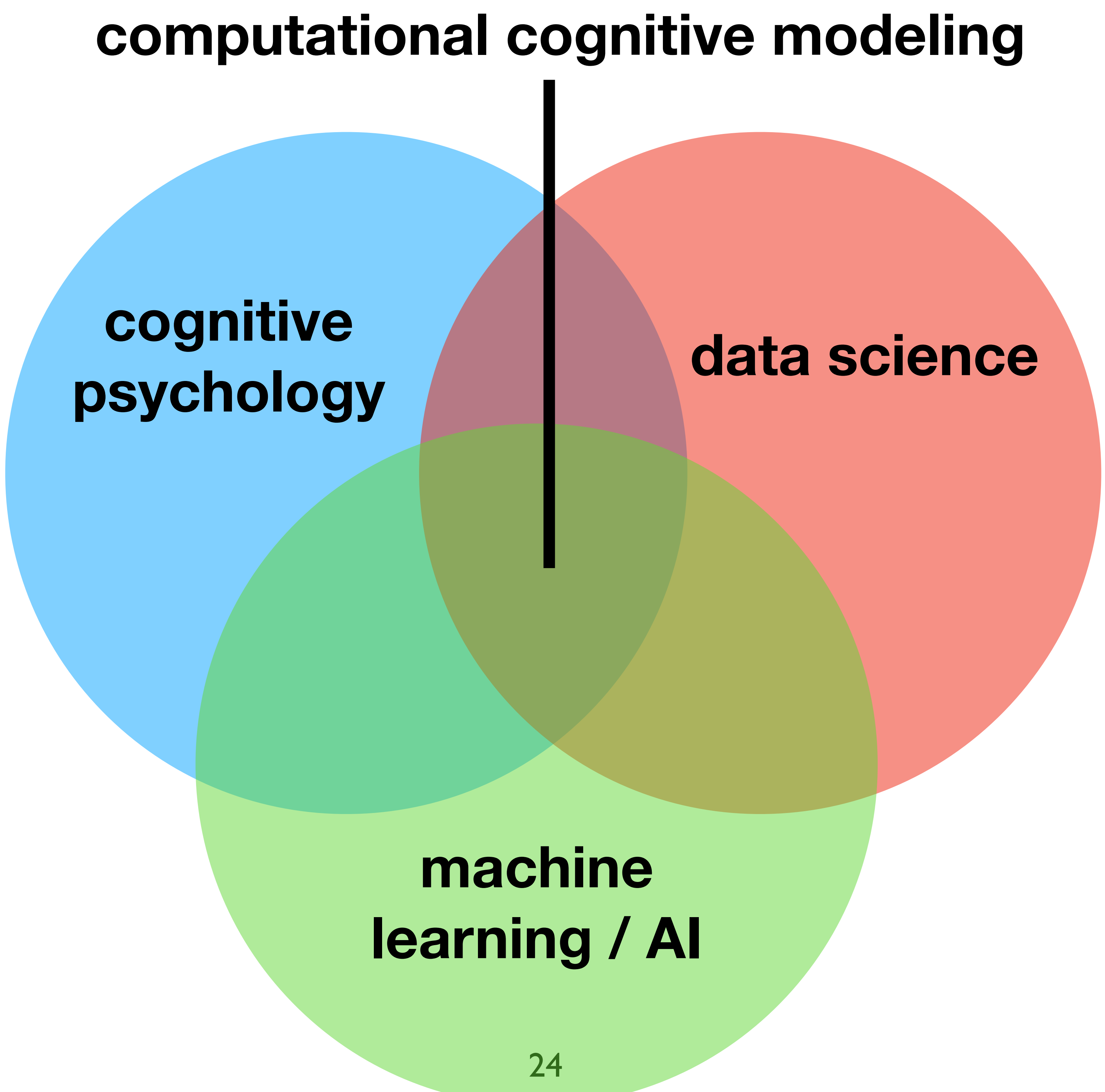
- Artificial neural networks
- Temporal difference learning
- Factor analysis
- Multi-dimensional scaling
- Probabilistic graphical models
- Structured Bayesian models
- Bayesian non-parametric models
- Probabilistic programming
- Recurrent neural networks
- …

Computational cognitive modeling can help make more powerful machines with more human-like learning capabilities

cognitive science / psychology → machine learning / AI / data science

cognitive science / psychology ← machine learning / AI / data science

Data science is about **extracting knowledge from data.** The human mind is the best general system we know of for **extracting knowledge from data**.



question asking compositional learning

one-shot learning scene understanding

concept learning transferring to new tasks

language acquisition inventing new tasks

computational problems that are easier for people than for machines

creativity

language understanding

Special opportunities for improving machine learning and AI through both engineering and REVERSE engineering.

general purpose problem solving

commonsense reasoning

self-assessment

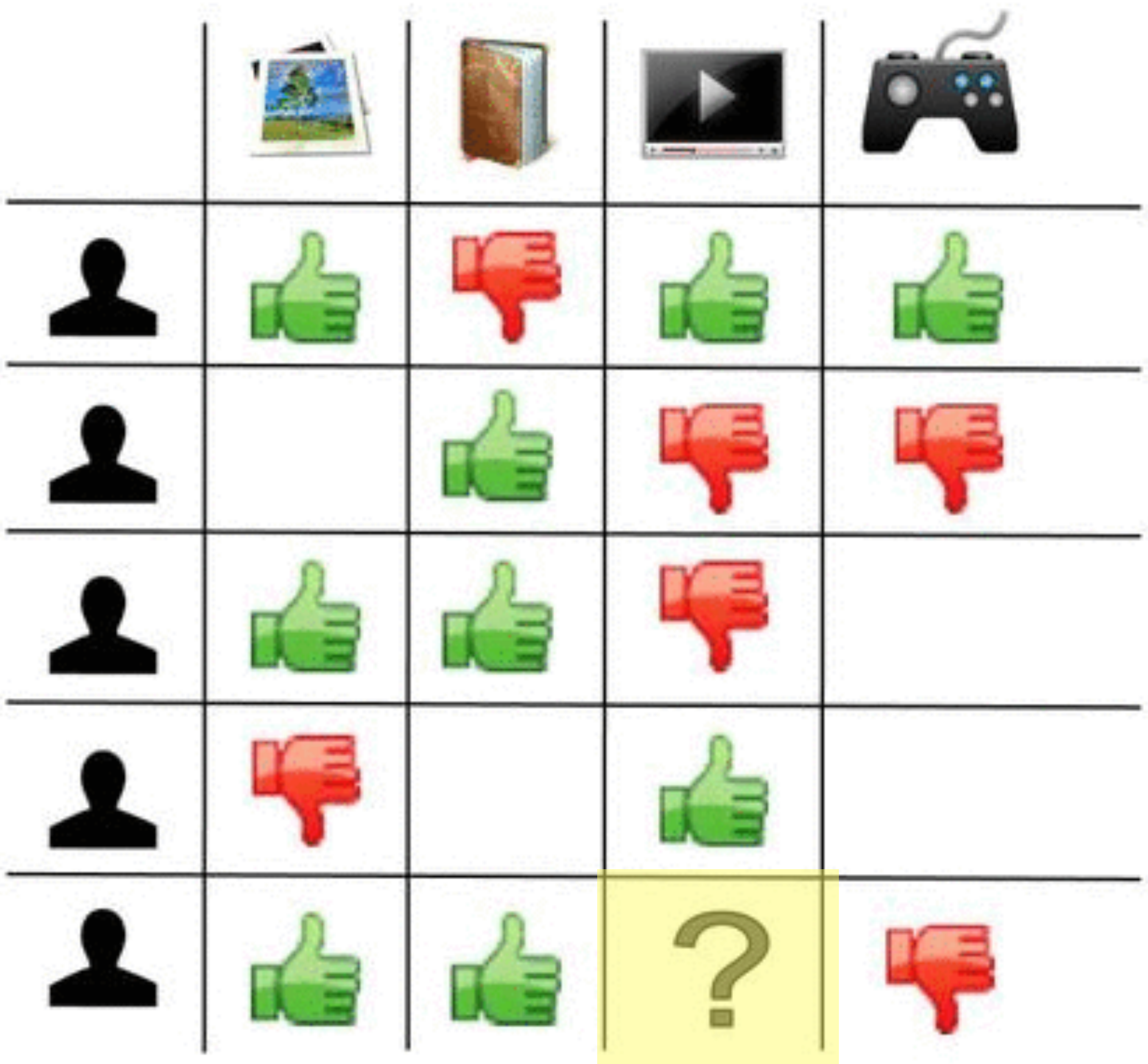
forming explanations curiosity and motivation

Can we better understand behavioral data by building computational cognitive models?

- In practice, data scientists deal with huge quantities of behavioral data..

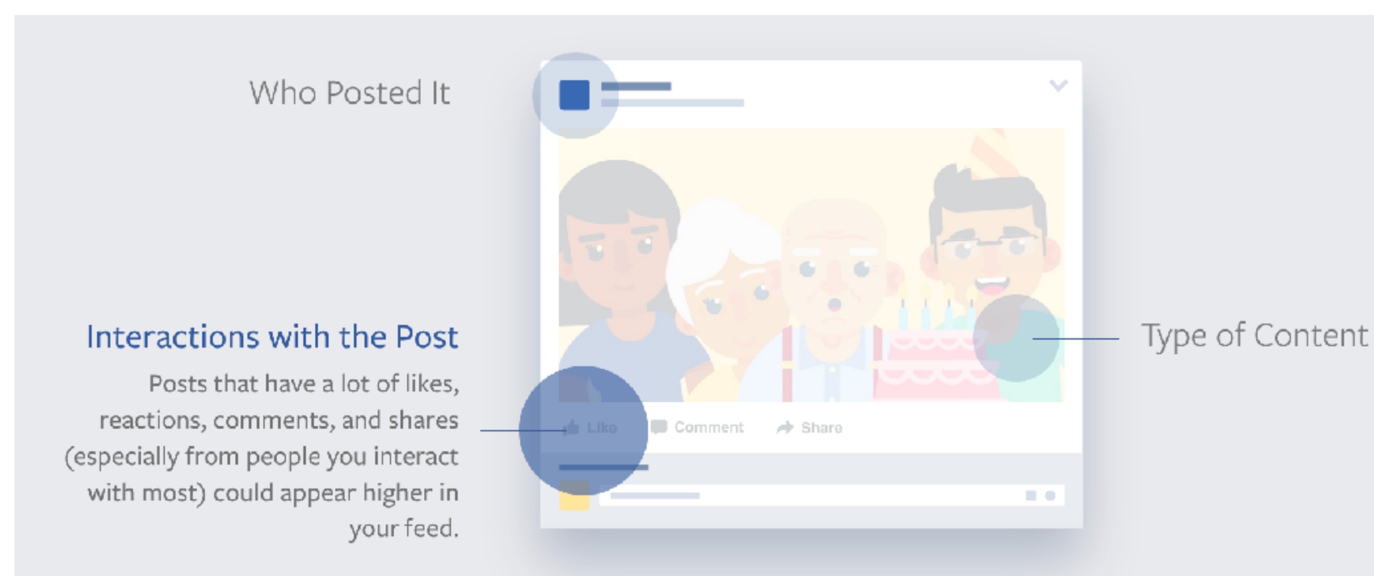
popular applications with behavioral data

collaborative filtering



churn modeling

adaptive content (e.g., news feed)

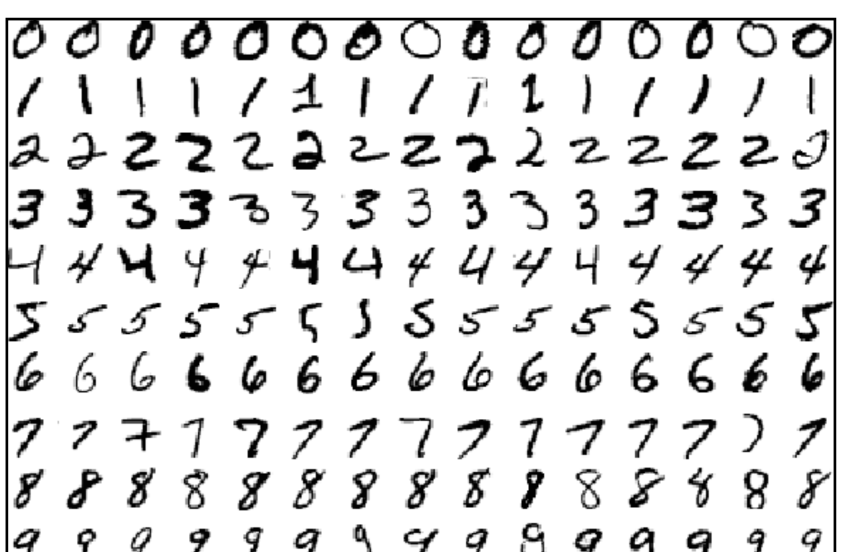


popular challenges for developing machine learning / AI algorithms

object recognition (ImageNet)

caption generation (MSCOCO)

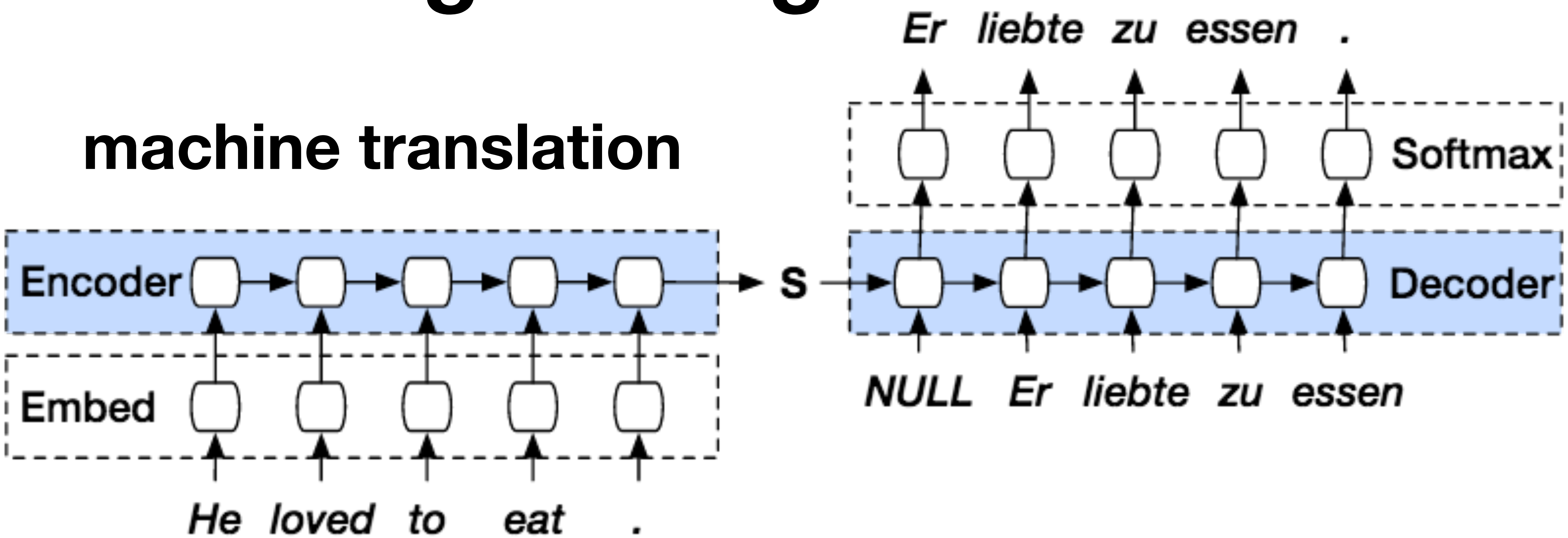
digit recognition (MNIST)



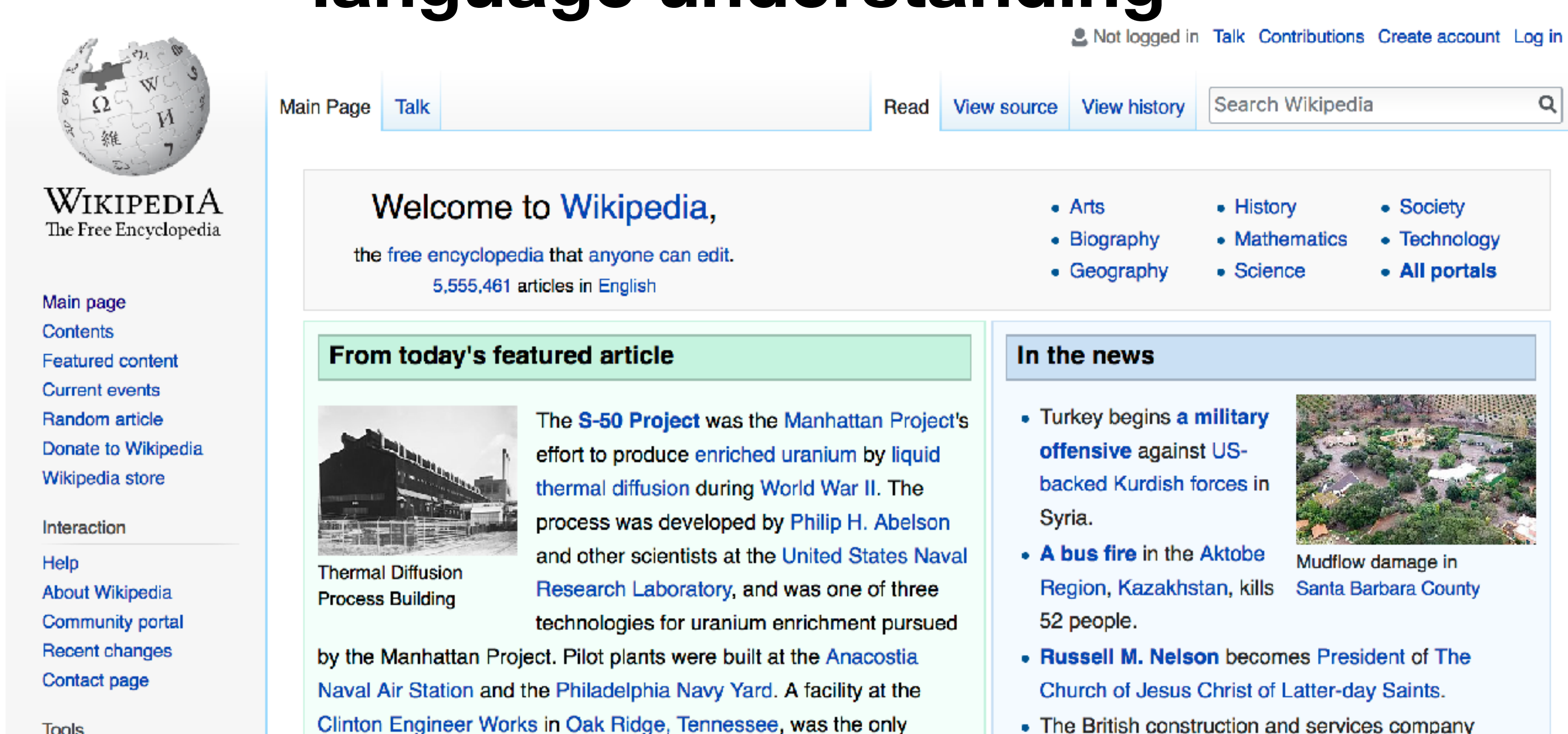
- Datasets consist of photos taken by PEOPLE, or of digits actually drawn by PEOPLE
- Task is to predict labels and sentences produced by PEOPLE, identifying objects and events that are meaningful to PEOPLE. In many cases the labels identify concepts invented by PEOPLE

popular challenges for developing machine learning / AI algorithms

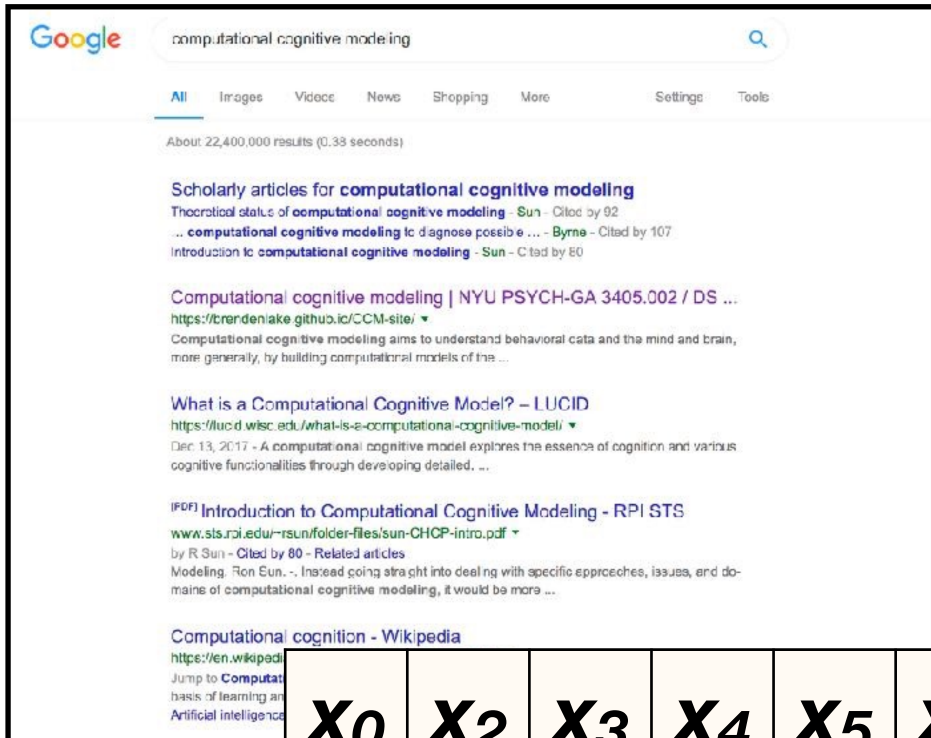
machine translation



language modeling and natural language understanding



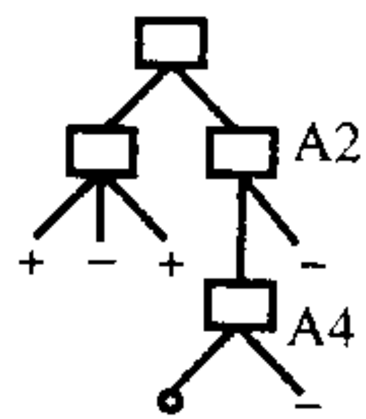
positing a mind to explain and predict behavior



x_0	x_2	x_3	x_4	x_5	x_6	x_7
0	0	0	1	0	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
1	1	1	1	0	1	1

rather than trying to predict clicks directly from browser history…

$p(y|x;\theta)$

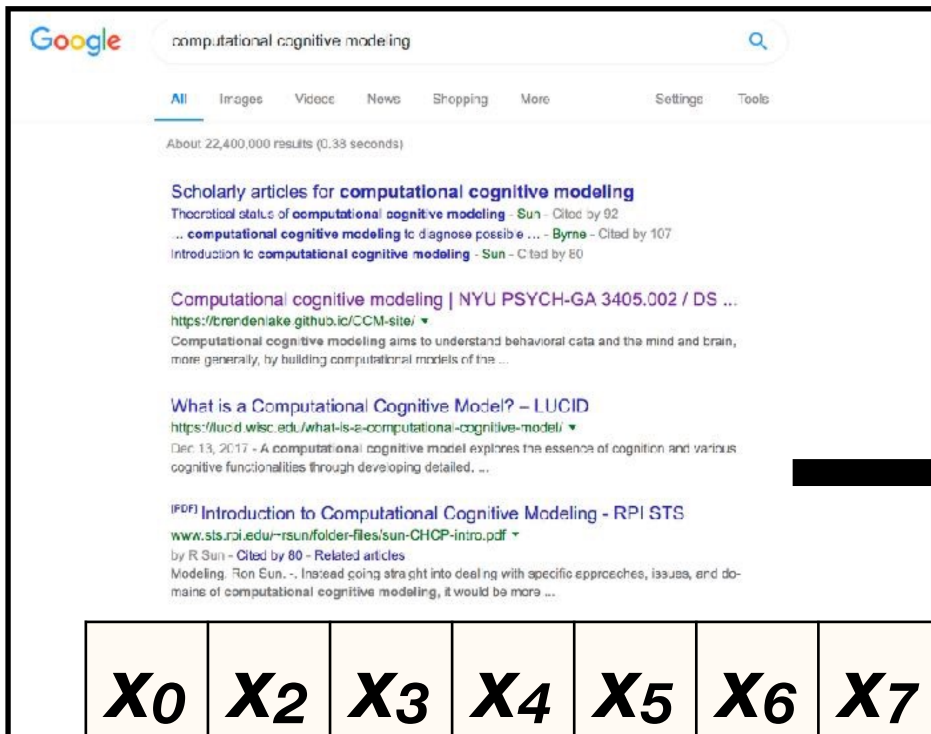


y
0
0
1
1

see Griffiths (2014). Manifesto for a new (computational) cognitive revolution.

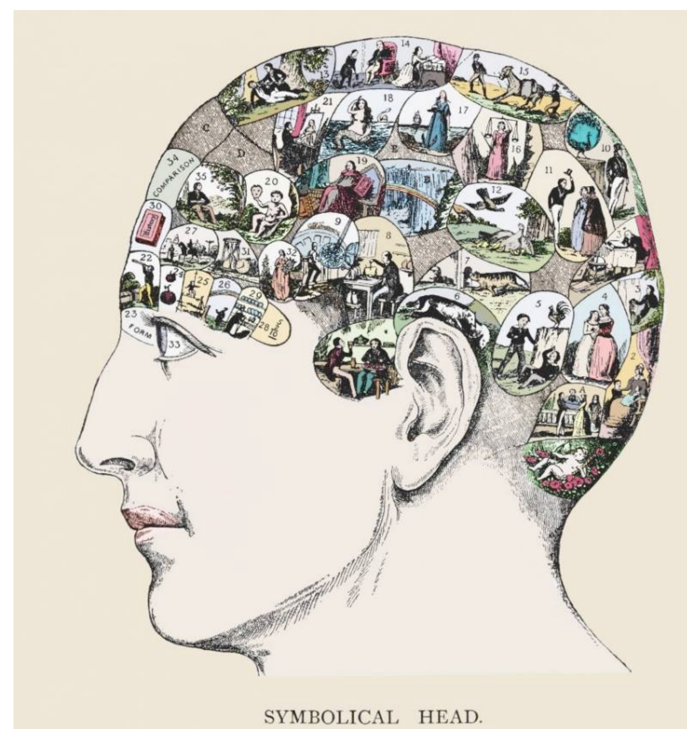
positing a mind to explain and predict behavior

- This course aims to show the value of positing mental processes to explain and predict behavior, and that mental processes are readily modeled with familiar computational tools to a data scientist.
- **Important caveat: This perspective is not yet mainstream in data science. This course is will teach you the right tools, but it's up to you to make the connections to practice!**



x_0	x_2	x_3	x_4	x_5	x_6	x_7
0	0	0	1	0	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
1	1	1	1	0	1	1

computational cognitive modeling



y
0
0
1
1

see Griffiths (2014). Manifesto for a new (computational) cognitive revolution.

Critical connections with neuroscience also, but this class is about modeling **higher-level cognitive rather than neural processes**

cognitive science / psychology

machine learning / AI / data science

neuroscience

****not this class!**

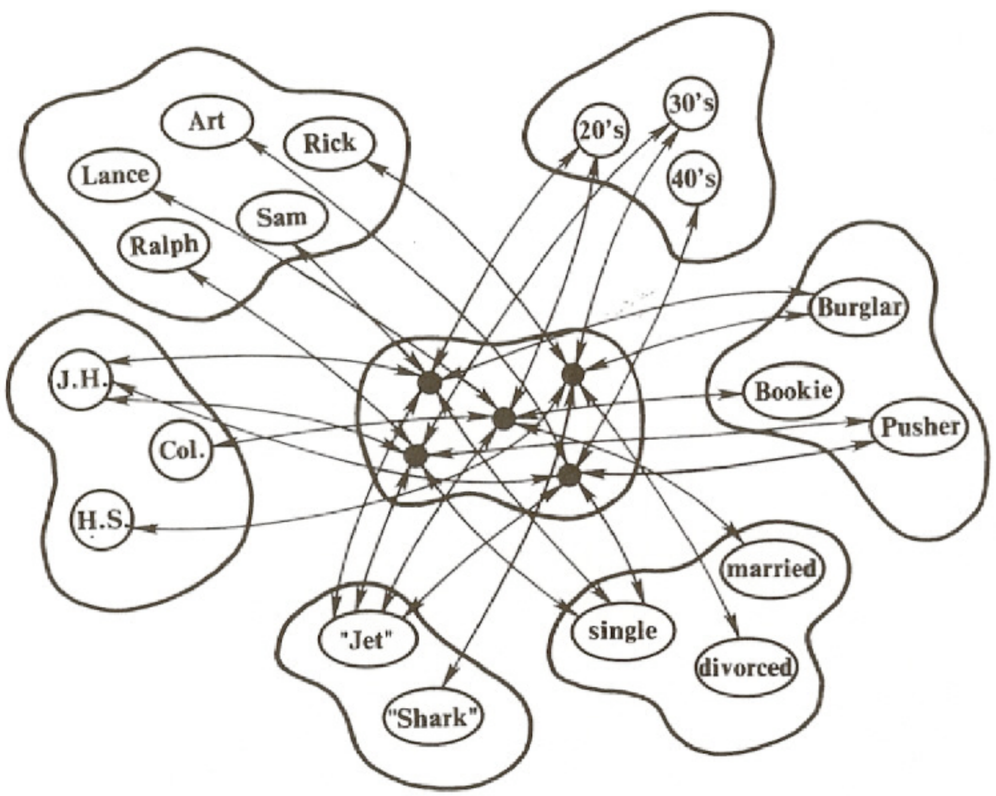
We will spend most of our time diving into various computational modeling paradigms

- Neural networks / deep learning
- Reinforcement learning
- Bayesian modeling
- Classification/categorization
- Probabilistic graphical models
- Program induction and language of thought models

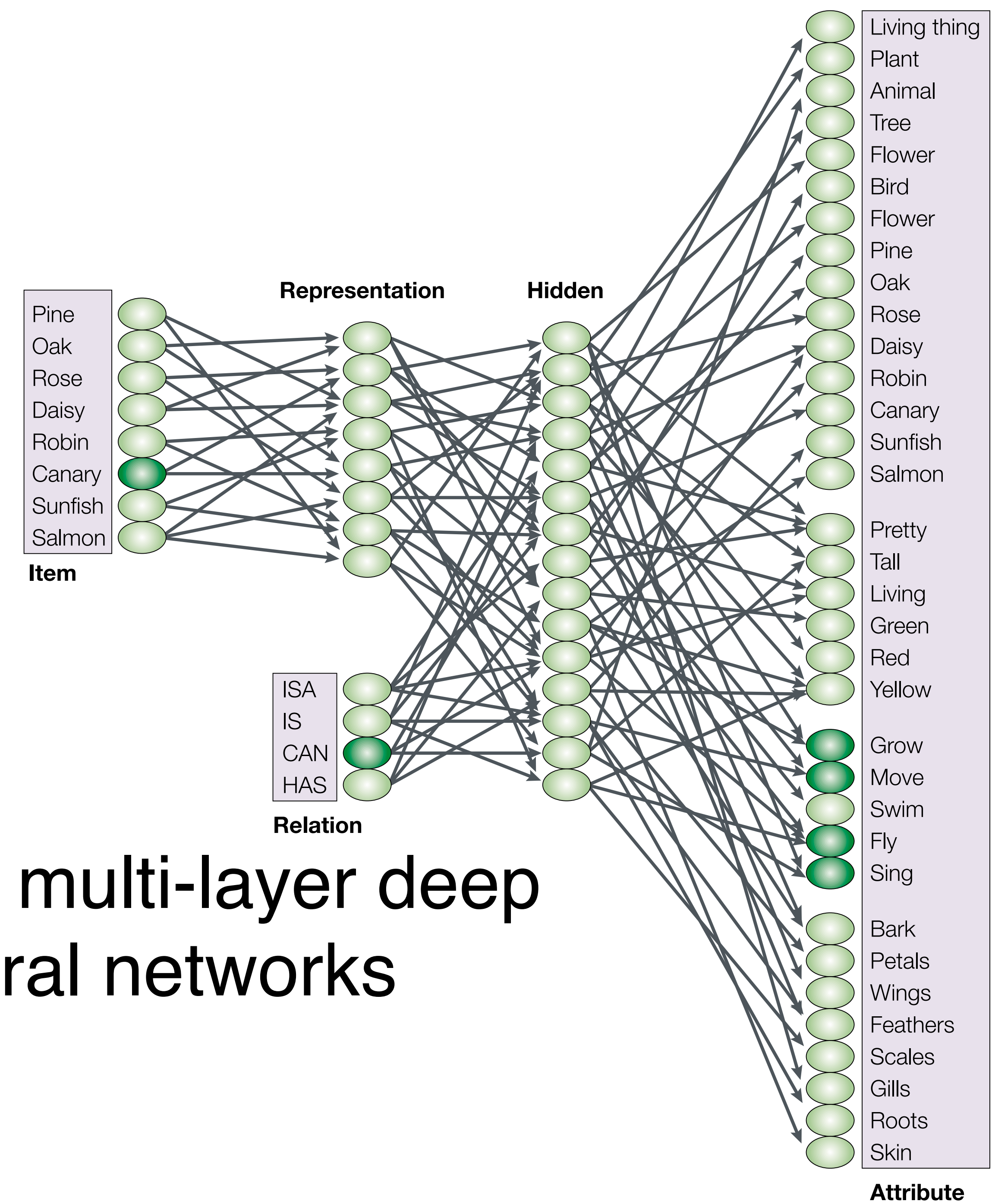
Notice synergy with contemporary machine learning / data science!

Neural networks / deep learning

Retrieving information from memory



Learning about objects and their properties; modeling cognitive development

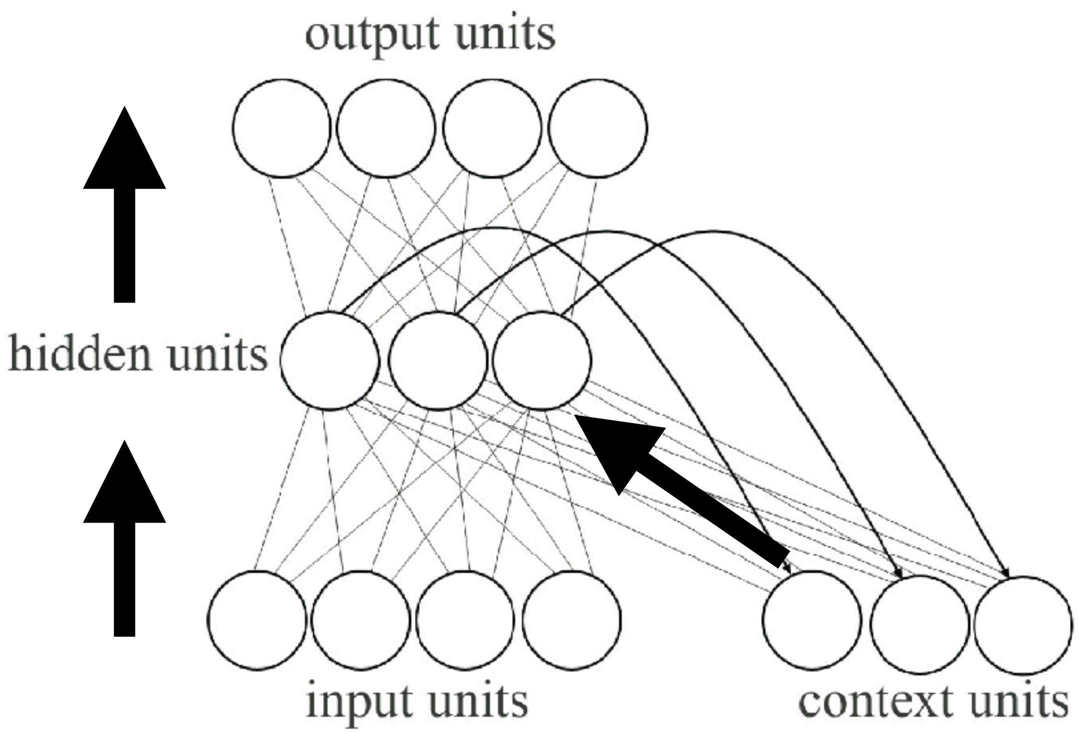


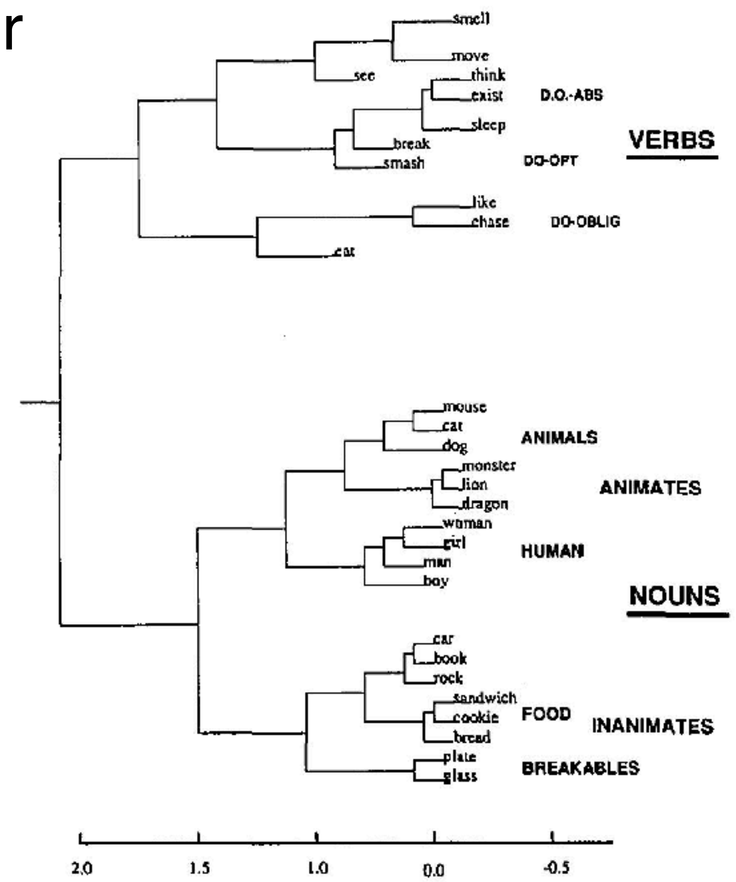
Training multi-layer deep neural networks

Neural networks / deep learning

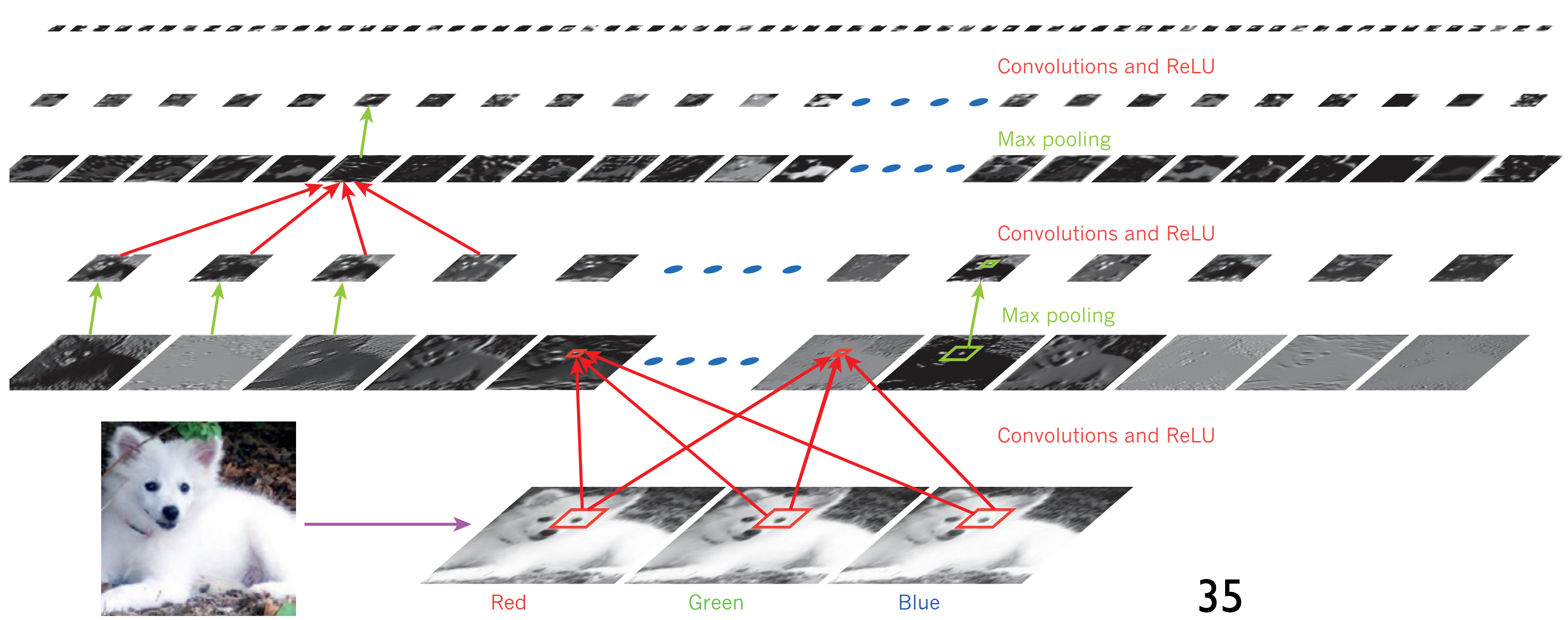
Recurrent neural networks

(Training RNNs with backpropagation was first done for computational cognitive modeling!)

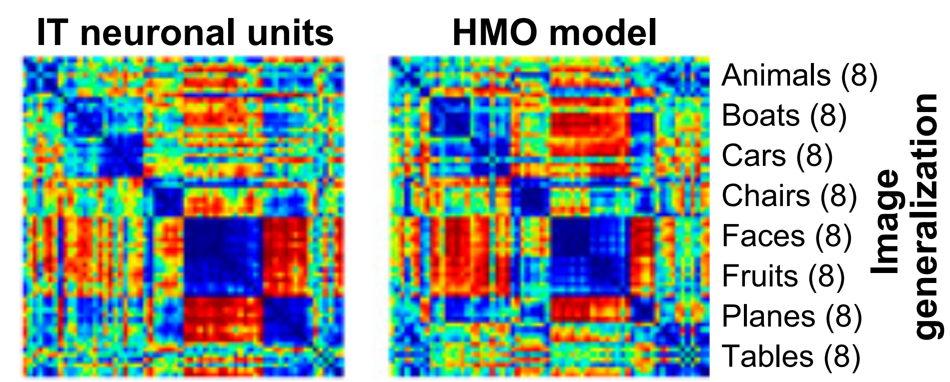




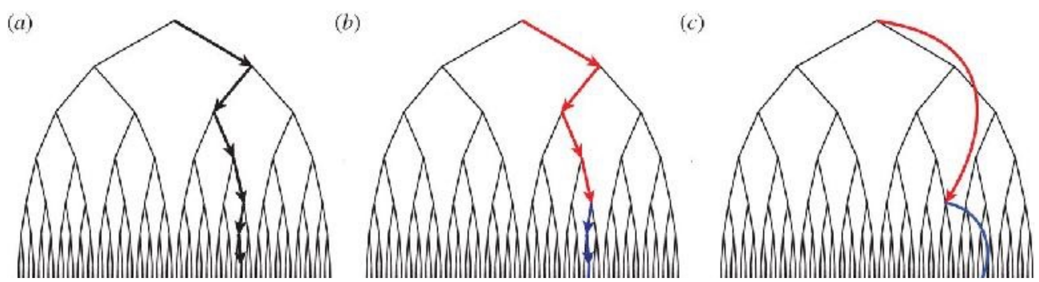
convolutional neural networks

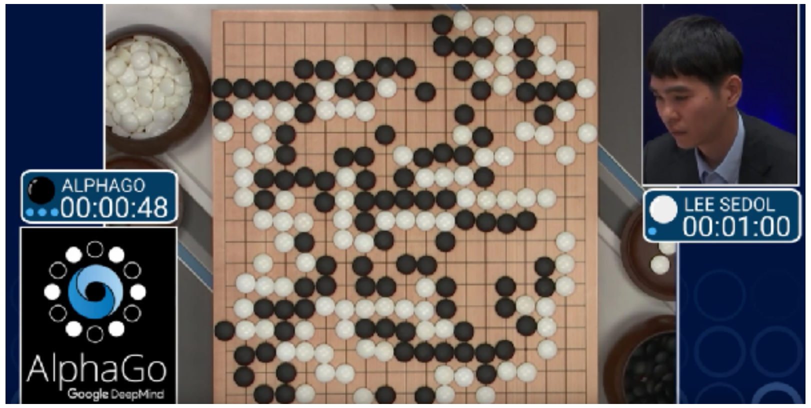


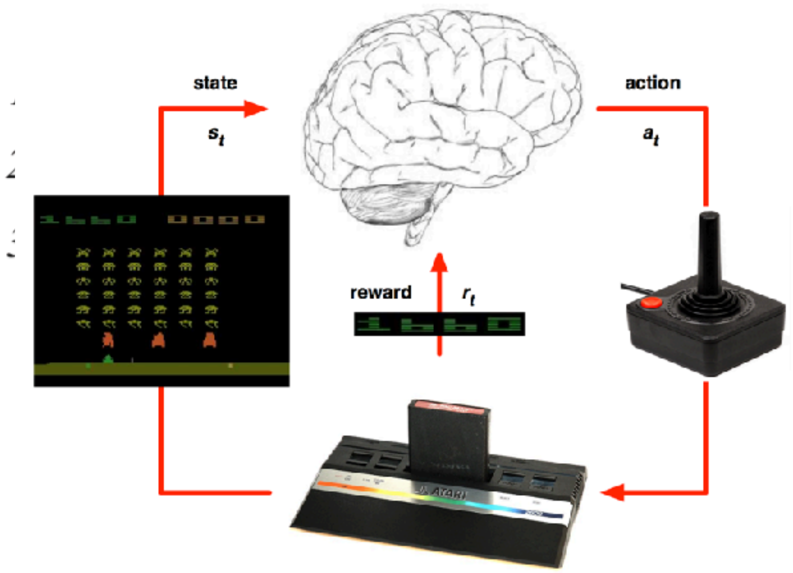
applications in cognitive science (and a bit of neuroscience)

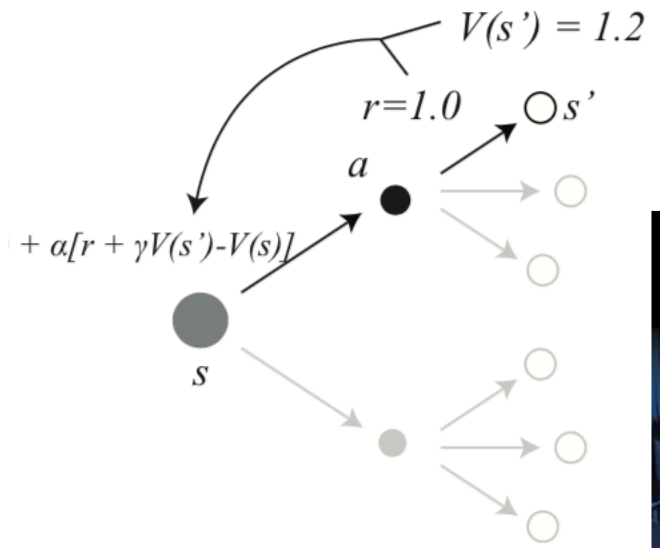


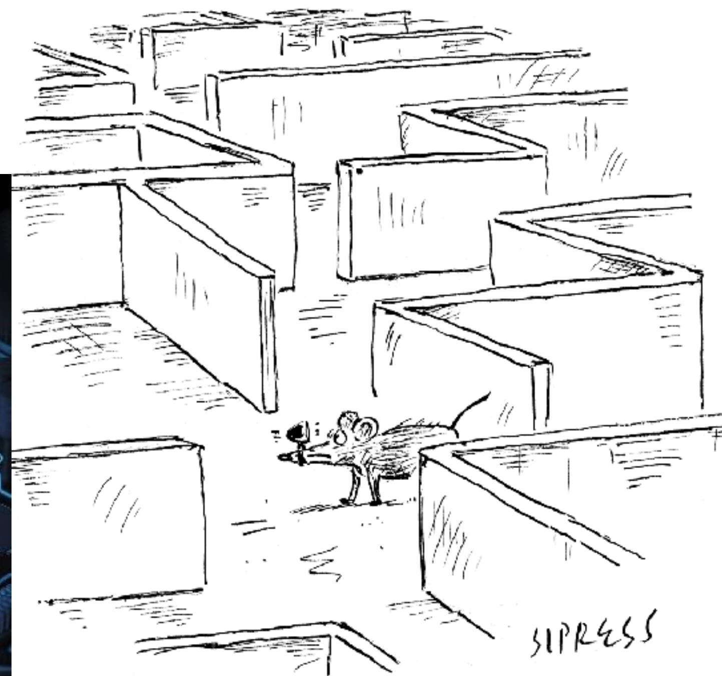
Reinforcement learning



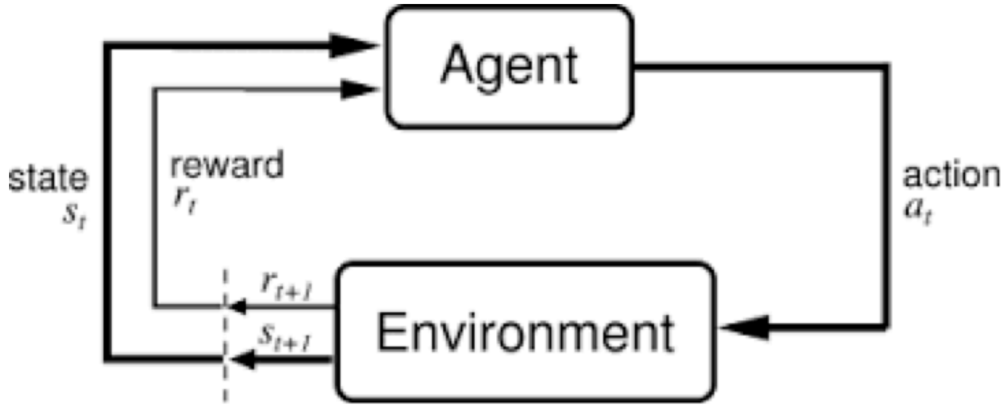


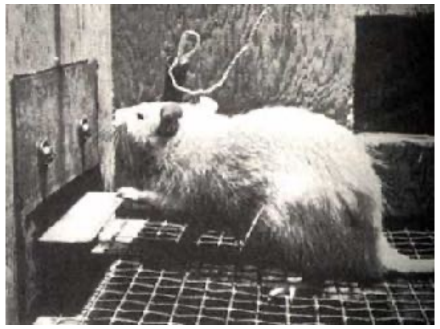






"Recalculating ... recalculating ..."





Bayesian modeling

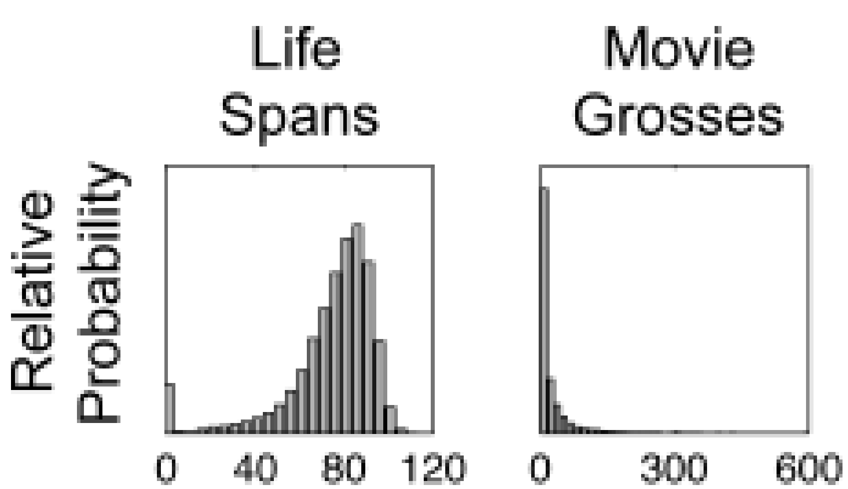
$$P(h|D) = \frac{P(h)P(D|h)}{\sum_{h_i} P(h_i)P(D|h_i)}$$

h : hypothesis $\quad D$: data

Predicting the future

You meet a man who is 75 years old. How long will he live?

A movie has grossed 75 million dollars at the box office, but you don't know how long it's been running. How much will it gross total?

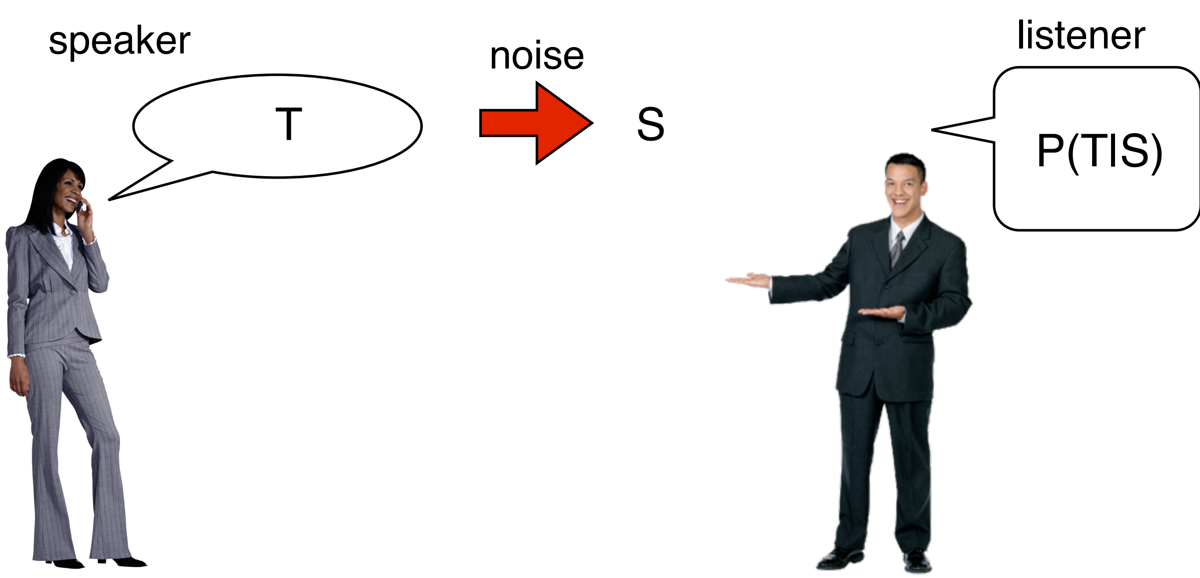


Property induction

Cows use biotin for hemoglobin synthesis
Seals use biotin for hemoglobin synthesis
—Therefore—
All mammals use biotin for hemoglobin synthesis

How strong is this inductive argument?

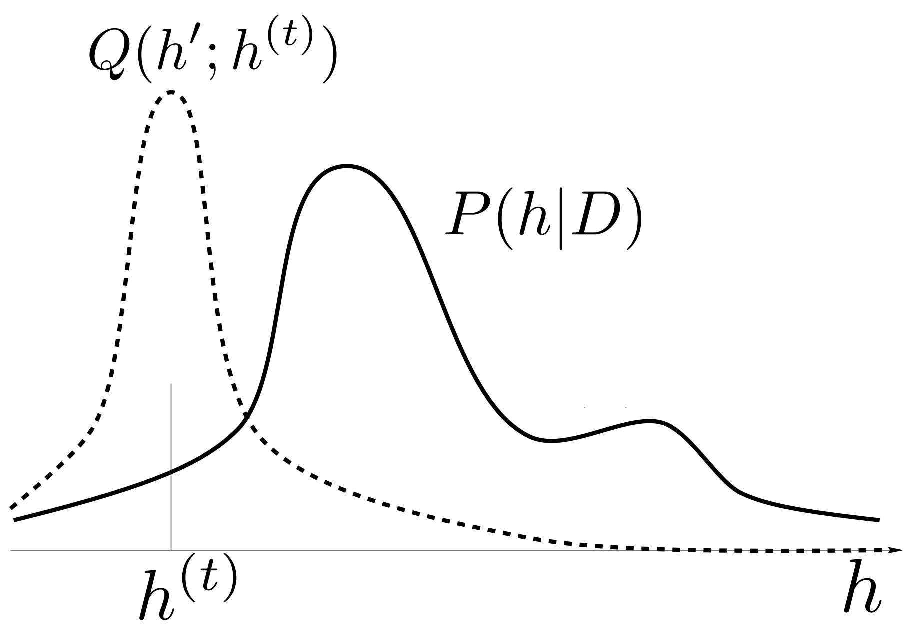
Speech perception under noise

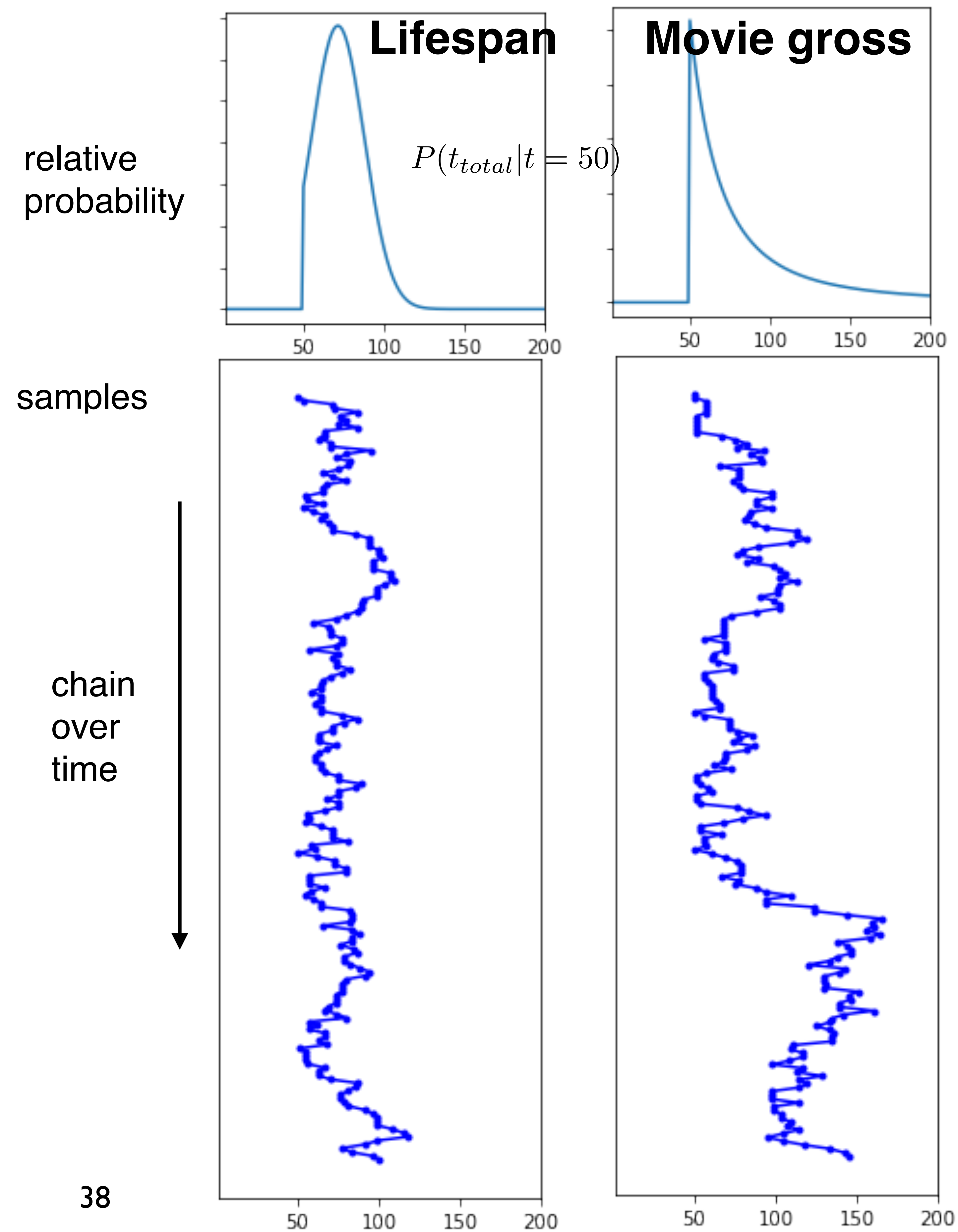


Inference in Bayesian models

- Exact inference
- Monte Carlo methods
  - Importance sampling
  - Markov Chain Monte Carlo

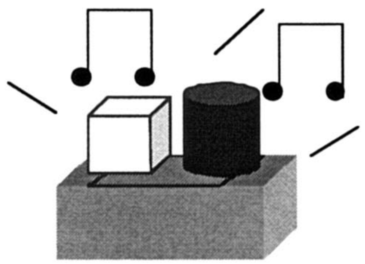
$$E[\phi(h)|D] \approx \frac{1}{T}\sum_t \phi(h^{(t)})$$



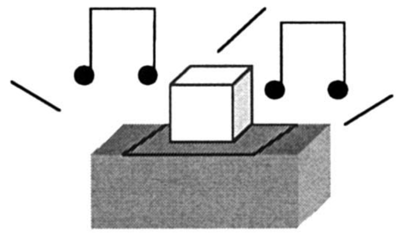


Probabilistic graphical models

Causal learning as structure learning

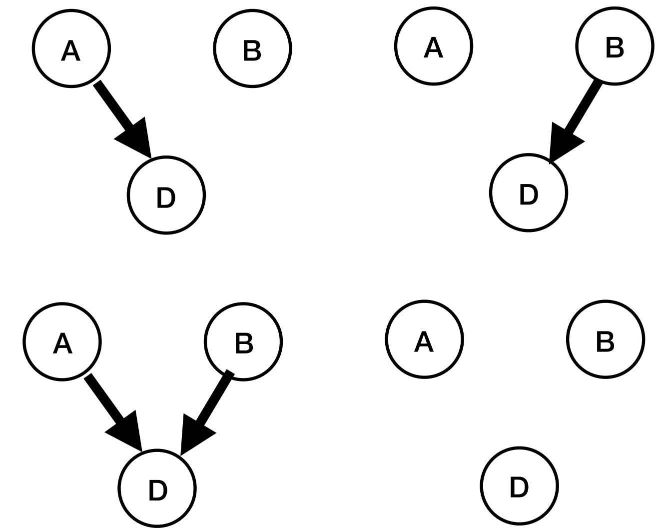


Both objects activate the detector

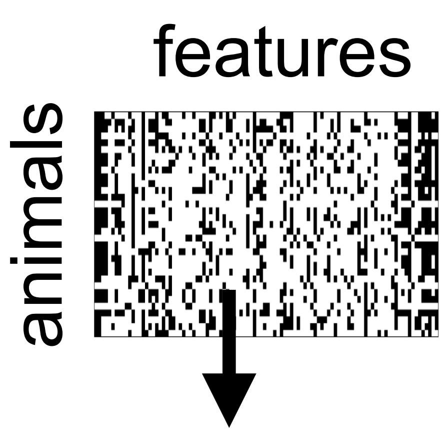


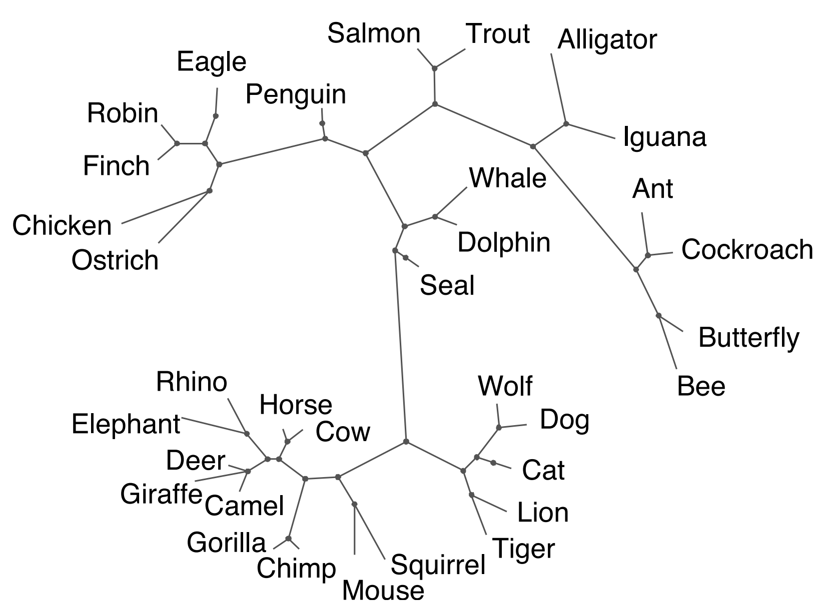
Object A activates the detector by itself

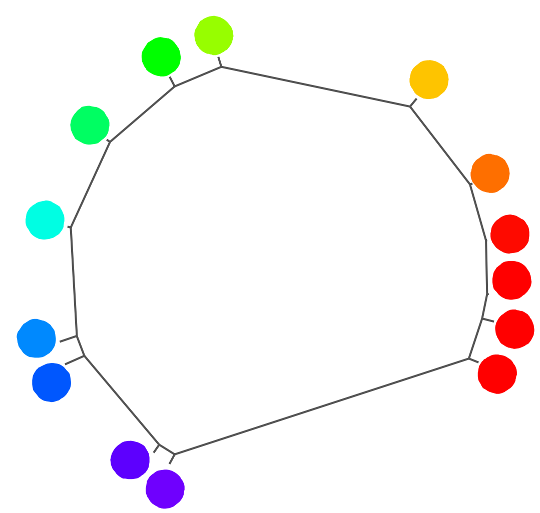
Children are asked if each is a blicket, then they are asked to make the machine go



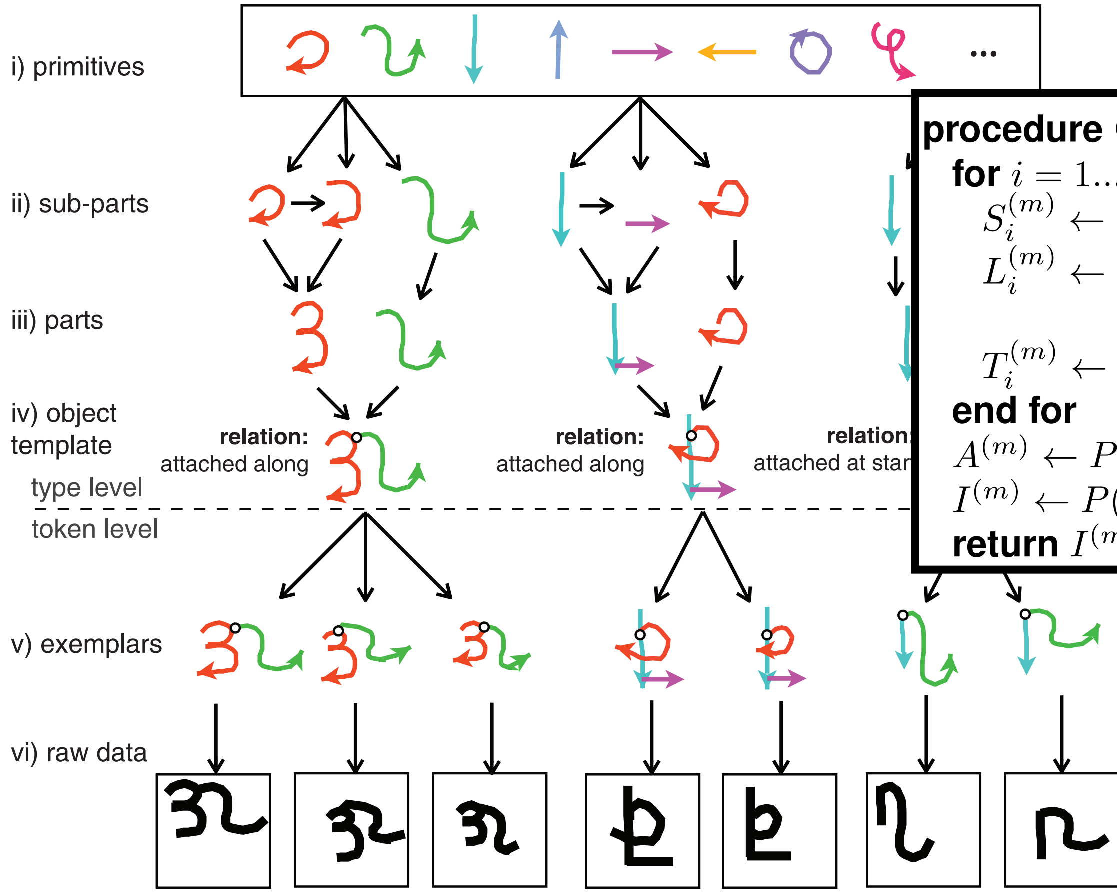
Structure discovery and evaluating inductive arguments







Program induction and language of thought models



procedure GENERATETOKEN(ψ)

  for $i = 1...\kappa$ **do**

    $S_i^{(m)} \leftarrow P(S_i^{(m)}|S_i)$ ▷ Add motor variance

    $L_i^{(m)} \leftarrow P(L_i^{(m)}|R_i, T_1^{(m)}, ..., T_{i-1}^{(m)})$ ▷ Sample part's start location

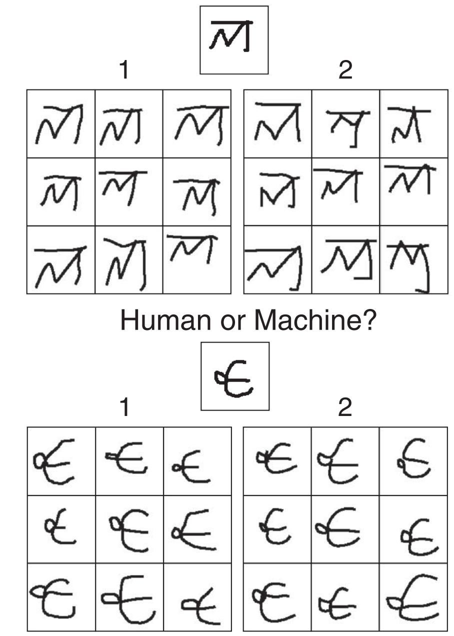
    $T_i^{(m)} \leftarrow f(L_i^{(m)}, S_i^{(m)})$ ▷ Compose a part's trajectory

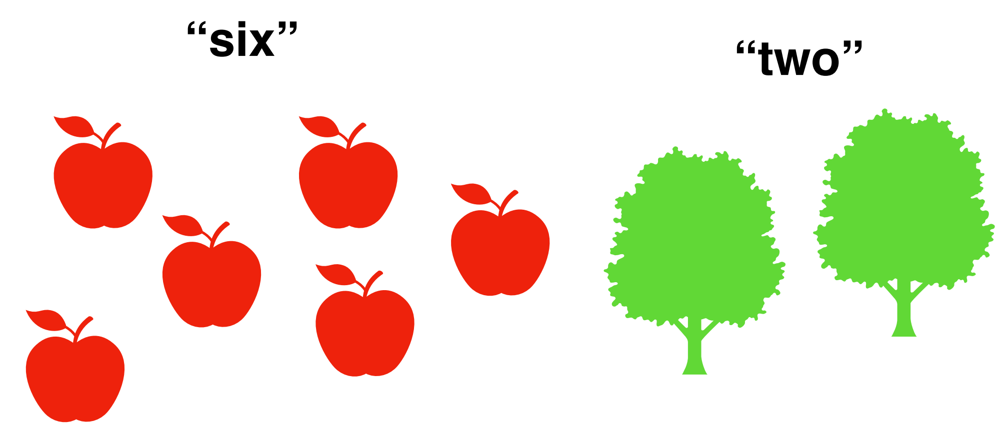
  end for

  $A^{(m)} \leftarrow P(A^{(m)})$ ▷ Sample affine transform

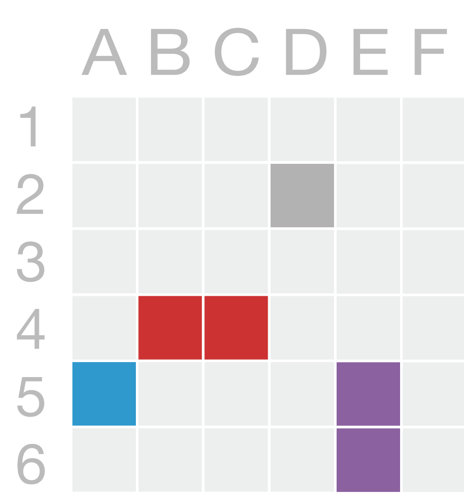
  $I^{(m)} \leftarrow P(I^{(m)}|T^{(m)}, A^{(m)})$ ▷ Sample image

  return $I^{(m)}$





λ S . (if (singleton? S)
"one"
(if (doubleton? S)
"two"
undef))



What is the top left of all the ship tiles?

```
(topleft (setDifference (set 1A ... 6F) (coloredTiles Water)))
```

Are all the ships horizontal?

```
(all (map (lambda x (== H (orient x))) (set Blue Red Purple)))
```

Are blue and purple ships touching and red and purple not touching (or vice versa)?

```
(== (touch Blue Purple) (not (touch Red Purple)))
```


40


LOGISTICS

Course website

https://brendenlake.github.io/CCM-site/

Computational cognitive modeling - Spring 2024

NYU PSYCH-GA 3405.004 / DS-GA 1016.003

View On GitHub

This project is maintained by brendenlake

Computational cognitive modeling - Spring 2024

Instructors: Brenden Lake and Todd Gureckis

Teaching Assistants: Solim LeGris and Cindy Luo

Meeting time and location:

Lecture. Lectures are on **Thursdays 10-11:40AM** in 12 Waverly Pl Room G08. There is no zoom or lecture capture; if you can't make it to class, you can email us to request last year's video (instructors-ccm-spring2024@googlegroups.com).

Labs. Fridays 12:30-1:20PM in 12 Waverly Pl Room G08. Labs are recorded.

Brightspace access for waitlist and auditors. We won't need bright space for anything except lab recordings. If you feel you need bright space access and don't have it, please add your email to this spreadsheet. We will add the emails from the spreadsheet periodically.

Course numbers:

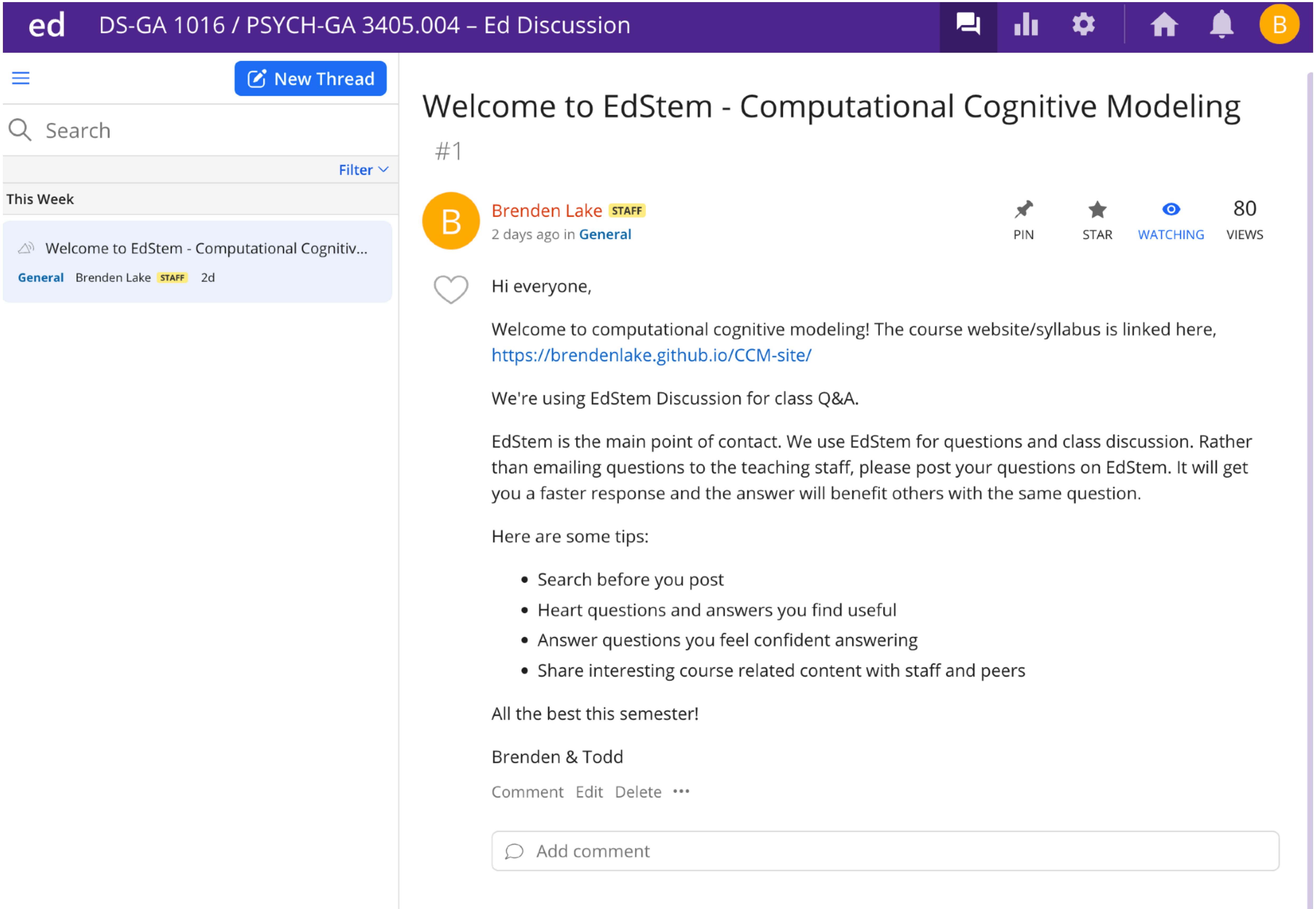
DS-GA 1016 (Data Science)

PSYCH-GA 3405.004 (Psychology)

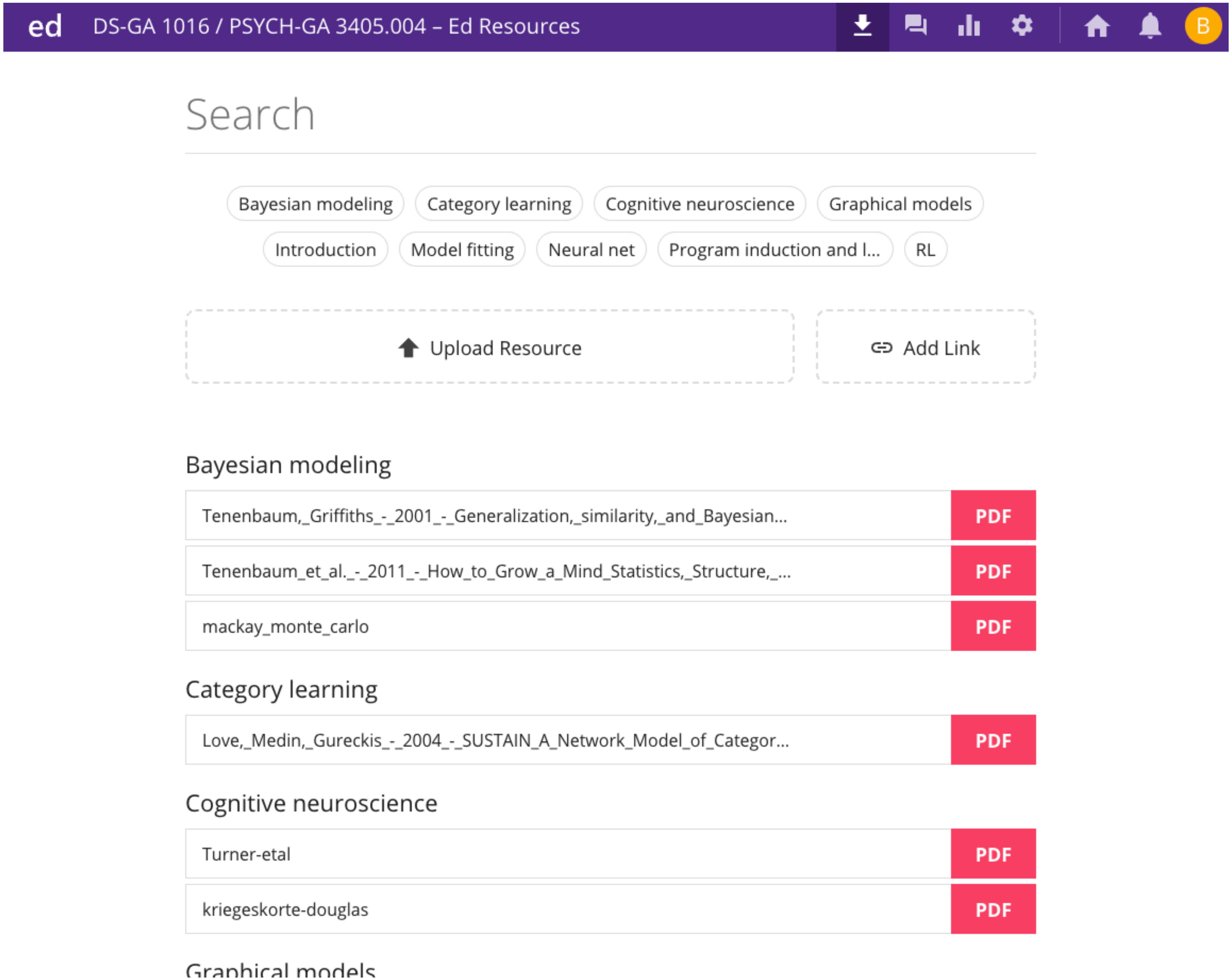
Contact information and Ed Discussion:

The class Ed Discussion page is the main point of contact. We use Ed Discussion for questions and class discussions. Rather than emailing questions to the teaching staff, please post your questions on Ed Discussion. It will get you a faster response and the answer will benefit others with the same question.

Course discussion: Ed Discussion



Readings posted on Ed Discussion



Getting in touch

Ed Discussion should be your main point of contact. If you have a question, and you think there is a possibility that someone may have the same question, please post it to EdStem for everyone's benefit. *Those registered for the course should be automatically enrolled, but there's a backup "join" link on the course website.*

If you need to send an individual message,

Email address for instructors and TAs:
instructors-ccm-spring2024@googlegroups.com

Lectures

Thursdays 10-11:40AM in 12 Waverly Pl Room G08.

There is no zoom or lecture capture; if you can't make it to class, you can email us to request last year's video. (instructors-ccm-spring2024@googlegroups.com)

Labs

Fridays 12:30-1:20PM in 12 Waverly Pl Room G08.

Labs should have working lecture capture. (But we may have technical difficulties in first week or so)

Brightspace (not used)

We won't use it for much, unless you want to watch recorded labs.

If needed, auditors and folks on the waitlist can get added to brightspace. Please add your email to spreadsheet linked on the class website.

Lecture schedule

Thurs. Jan 25: Introduction
Thurs. Feb. 1: Neural networks / Deep learning (part 1)
Thurs. Feb. 8: Neural networks / Deep learning (part 2)
Thurs. Feb. 15: Reinforcement learning (part 1)
Thurs. Feb. 22: Reinforcement learning (part 2)
Thurs. Feb. 29: Reinforcement learning (part 3)
Thurs. Mar 7: Bayesian modeling (part 1)
Thurs. Mar 14: Bayesian modeling (part 2)(same slides as part 1)
Thurs. Mar. 21: No class, Spring break
Thurs. Mar. 28: Model comparison and fitting, tricks of the trade
Thurs. Apr 4: Categorization
Thurs. Apr 11: Probabilistic Graphical models
Thurs. Apr 18: Information sampling and active learning
Thurs. April 25: Program induction and language of thought models
Thurs. May 2: Computational Cognitive Neuroscience

Lab schedule

Fri. Jan 26, Python and Jupyter notebooks review
Fri. Feb 2, Introduction to PyTorch
Fri. Feb 9, HW 1 Review
Fri. Feb 16, No lab
Fri. Feb 23, Reinforcement learning
Fri. Mar 1, HW 2 review
Fri. Mar 8, Probability Review
Fri. Mar 15, HW 3 Review
Fri. Mar 22, No lab (Spring break)
Fri. Mar 29, No lab
Fri. Apr 5, TBD
Fri. Apr 12, HW 4 Review
Fri. Apr 19, TBD
Fri. Apr 26, TBD
Fri. May 3, TBD

Pre-requisites

- *Math:* We will use concepts from linear algebra, calculus, and probability. If you had linear algebra and calculus as an undergrad, or if you have taken Math Tools in the psychology department, you will be in a good position for approaching the material. Familiarity with probability is also assumed. We will review some of the basic technical concepts in lab.

- *Programming:* Previous experience with Python is required. Previous IN CLASS experience with Python is strongly recommend—it's assumed you know how to program in Python. The assignments will use Python 3 and Jupyter Notebooks (http://jupyter.org)

Grading:

- The final grade is based on the homeworks (65%) and the final project (35%). Class participation may be used in cases in borderline grades.

Final project:

- The final project will be done in groups of 3-4 students. A short paper will be turned in describing the project (approximately 6 pages). The project will represent either an substantial extension of one of the homeworks (e.g., exploring some new aspect of one of the assignments), implementing and extending an existing cognitive modeling paper, or a cognitive modeling project related to your research. We provide a list of project ideas (see website), but of course you do not have to choose from this list.

Homeworks — programming requirements

Programming: We assume you are familiar with programming in Python

Homeworks use this setup:

- Python 3
- Jupyter notebooks
- Standard Python packages for scientific computing
    - numpy
    - scipy
    - pandas
    - matplotlib
- PyTorch library for neural networks

Using your laptop setup is encouraged!

Jupyter notebooks

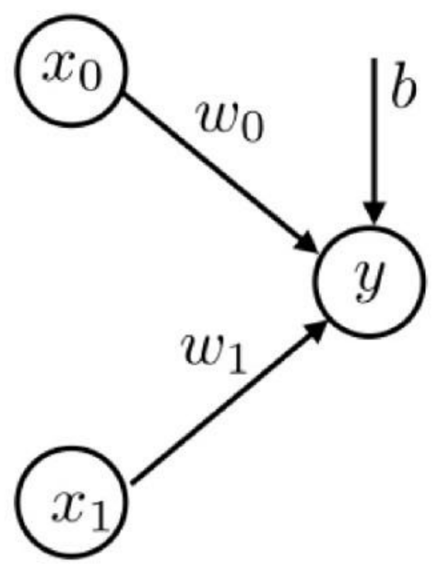
Homework - Neural networks - Part B (20 points)

Gradient descent for an artifical neuron

by *Brenden Lake* and *Todd Gureckis*
Computational Cognitive Modeling
NYU class webpage: https://brendenlake.github.io/CCM-site/
email to course instructors: instructors-ccm-spring2019@nyuccl.org

This homework is due before midnight on Monday, Feb. 25, 2019.

This assignment implements the gradient descent algorithm for a simple artificial neuron. As covered in lecture, the neuron will learn to compute logical OR. The neuron model and logical OR are shown below, for inputs x_0 and x_1 and target output y.



logical OR

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	1

This assignment requires some basic PyTorch skills, which were covered in lab. You can also review two basic PyTorch tutorials, "What is PyTorch?" and "Autograd", which have the basics you need.

In [ ]:
```
# Import libraries
from __future__ import print_function
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import torch
import torch.nn as nn
```

Let's create `torch.tensor` objects for representing the data matrix `D` with targets `Y`. Each row of `D` is a different data point.

In [ ]:
```
# Data
D = np.zeros((4,2),dtype=float)
D[0,:] = [0.,0.]
D[1,:] = [0.,1.]
D[2,:] = [1.,0.]
D[3,:] = [1.,1.]
```

Pre-configured cloud environment

Students registered for the course have the option of completing homework assignments on their personal computers (encouraged if know how to set it up!), or in a cloud Jupyter environment with all required packages pre-installed (see website).

Collaboration and honor code

We take the collaboration policy and academic integrity **very seriously**. Violations of the policy will result in zero points and possible disciplinary referral.

You may discuss the homework assignments with your classmates, but you must run the simulations and complete the write-ups for the homeworks on your own. **Under no circumstance should students look at each other's code or write ups, or code/write-ups from previous years of this course.** Do not share your write up or code with any of your classmates under any circumstances.

Course policies

Late work:

- We will take off 10% for each day a homework or final project is late.

See policy on extensions, regrading, no extra credit, etc. on syllabus

Laptops in class:

- Laptops in class are discouraged. We know many try to take notes on their laptops, but it's easy to get distracted (social media, etc.). **This also distracts everyone behind you!**

We encourage you to engage with the class and material, and engage with us as the instructors. Ask questions!

All slides are posted so there is no need to copy things down, and paper notes are great too.

Background survey

- Currently enrolled in what type of program:
  - Psychology Ph.D.? Psychology Masters? Data Science Masters? DS Ph.D.? Other graduate program? Undergraduate?

- Previous coursework:
  - Cognitive Psychology? Programming? Probability, statistics, MathTools? Machine learning? AI? Deep learning?

- Who knows about:
  - Prototype vs. exemplar models?
  - Categorical perception?
  - Semantic networks?
  - Logistic regression?
  - Backpropagation algorithm?
  - Simple recurrent network?
  - Model-based vs. model-free reinforcement learning?
  - Bayes' rule?
  - Conditional independence?
  - Conjugate prior?
  - Metropolis-Hastings?
  - Explaining away?
  - Probabilistic programming?

What you will come away with…

1. Experience with the major paradigms for computational cognitive modeling

2. An introduction to key technical tools (in Python and Jupyter notebooks):

- Neural networks / deep learning (in PyTorch)
- Reinforcement learning
- Bayesian modeling
- Model comparison and fitting
- Probabilistic graphical models
- Program induction and language of thought models

3. How to build computational models to test and evaluate psychological theories, and to understand behavioral data by modeling the underlying cognitive processes.

4. Ideally, students will leave the course with a richer understanding of how computational modeling advances cognitive science, and how computational cognitive modeling can inform research in data science, machine learning, and artificial intelligence

Is this course a substitute for machine learning?

- **No. It's not a substitute, it's complementary.**
- This course does survey various computational paradigms (deep learning, reinforcement learning, Bayesian modeling, classification, graphical models, etc.), and there is some overlap with ML classes in terms of technical content.
- But unlike ML classes, this is also a cognitive science class. **Our examples and applications aim to understand human learning, reasoning, and development, and to understand intelligent behavior more generally.**
- We get into some mathematical background, but ML courses take a more formal approach than we do here. We aim for a more accessible introduction.
- You will get hands on experience with running and analyzing complex models, implementing models, and analyzing behavioral data with computational models. Extensive final project.

For next time....

Readings for the next two lectures

- McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. The Appeal of Parallel Distributed Processing. Vol I, Ch 1.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. Nature 521:436–44.
- McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. Nature Reviews Neuroscience, 4(4), 310-322.
- Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179-211.
- Peterson, J., Abbott, J., & Griffiths, T. (2016). Adapting Deep Network Features to Capture Psychological Representations. Presented at the 38th Annual Conference of the Cognitive Science Society.

Homework 1 on neural networks will be released before next class

Questions?