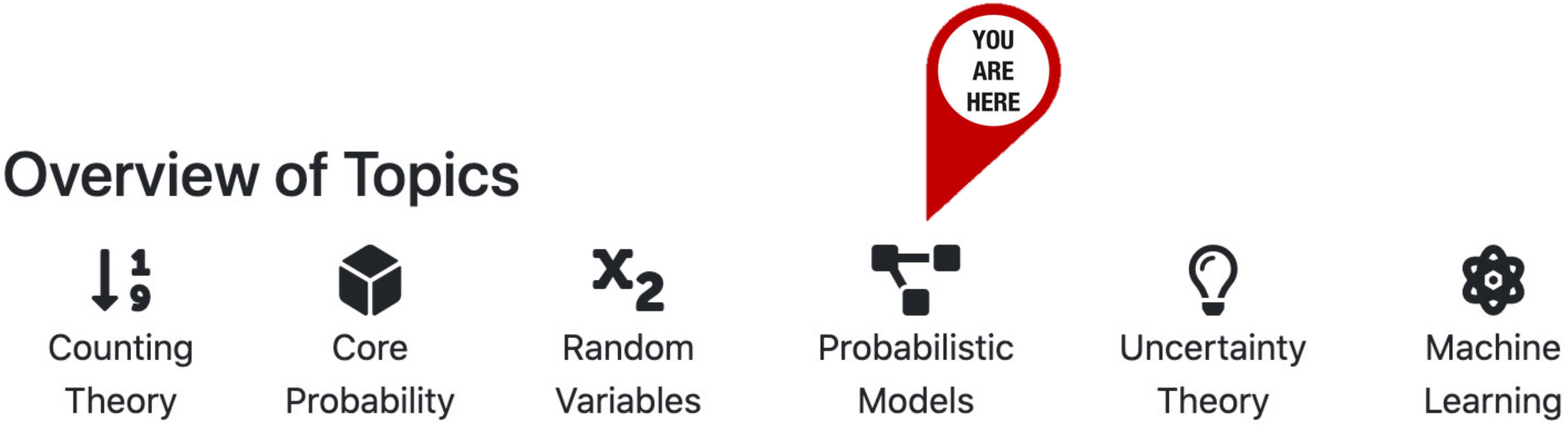
General Inference

Chris Piech
CS109, Stanford University

Announcements

- Midterm!!!!
- Most of you have finished the midterm! Some folks have not, so please don't talk about it with friends until Saturday.
- Pset 4 coming soon!!!

Where are we in CS109?



Where are we locally?

Discrete Models: General Case, Multinomial

Inference Conclusions from Observations

Modelling: Make your own!

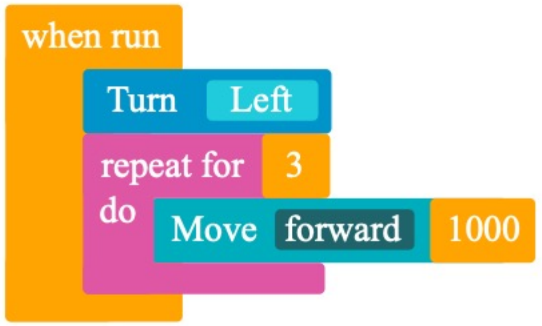
General Inference: Use computers to infer

Where do models come from?

Computers Couldn't Understand Code

60,000 students attempted this problem
37,000 unique solutions

Challenge

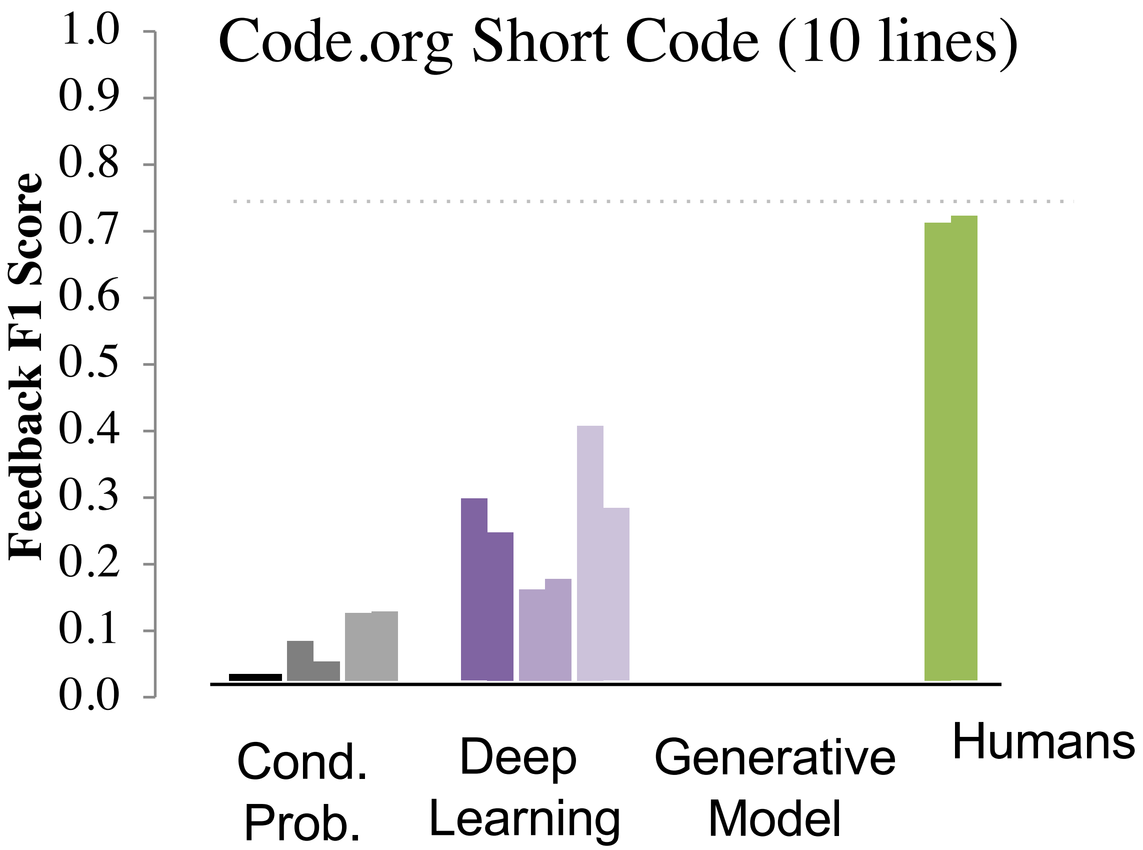


Student Code

You need to move and turn in your loop

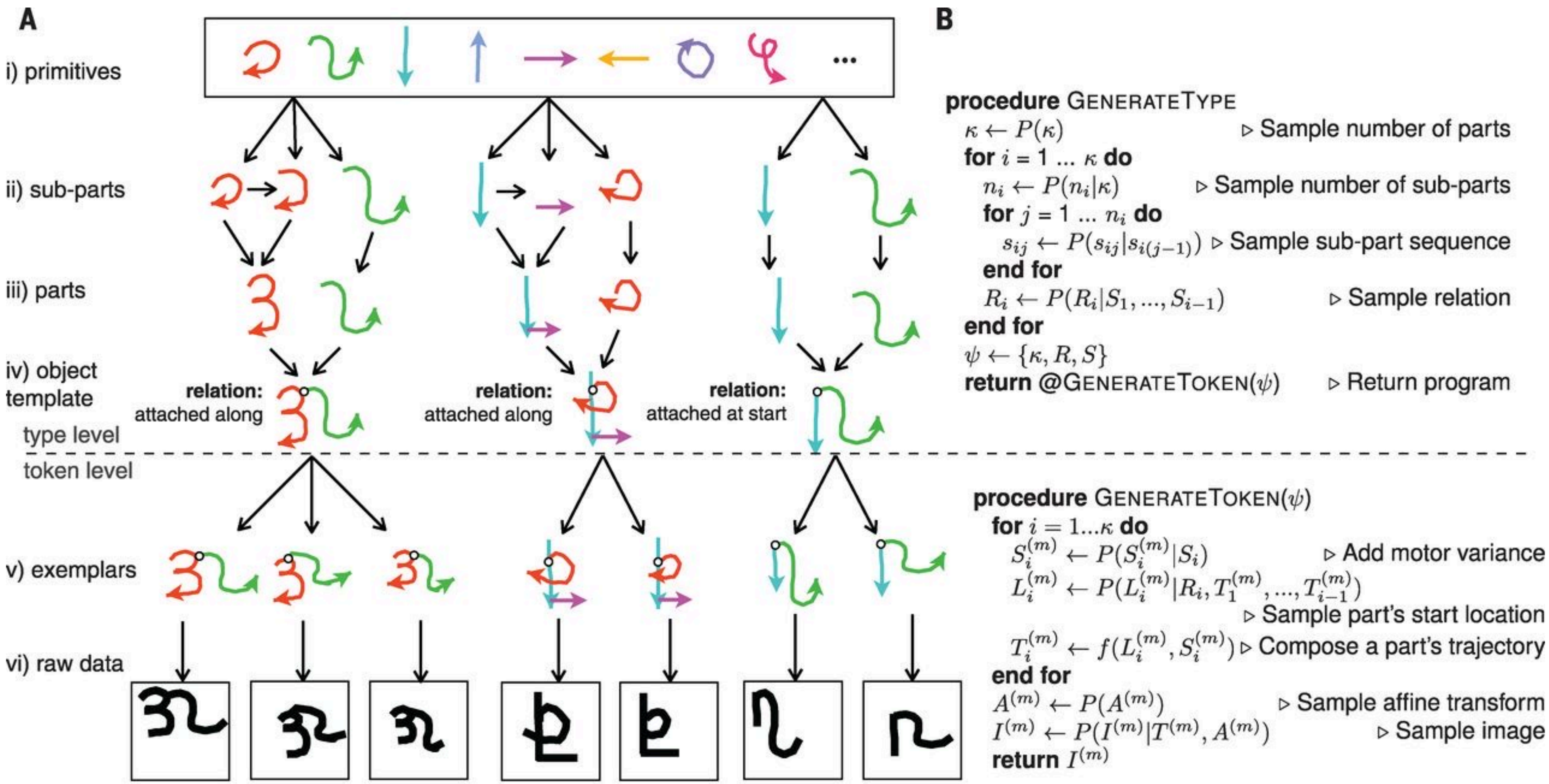
Insight

Computers Couldn't Understand Code

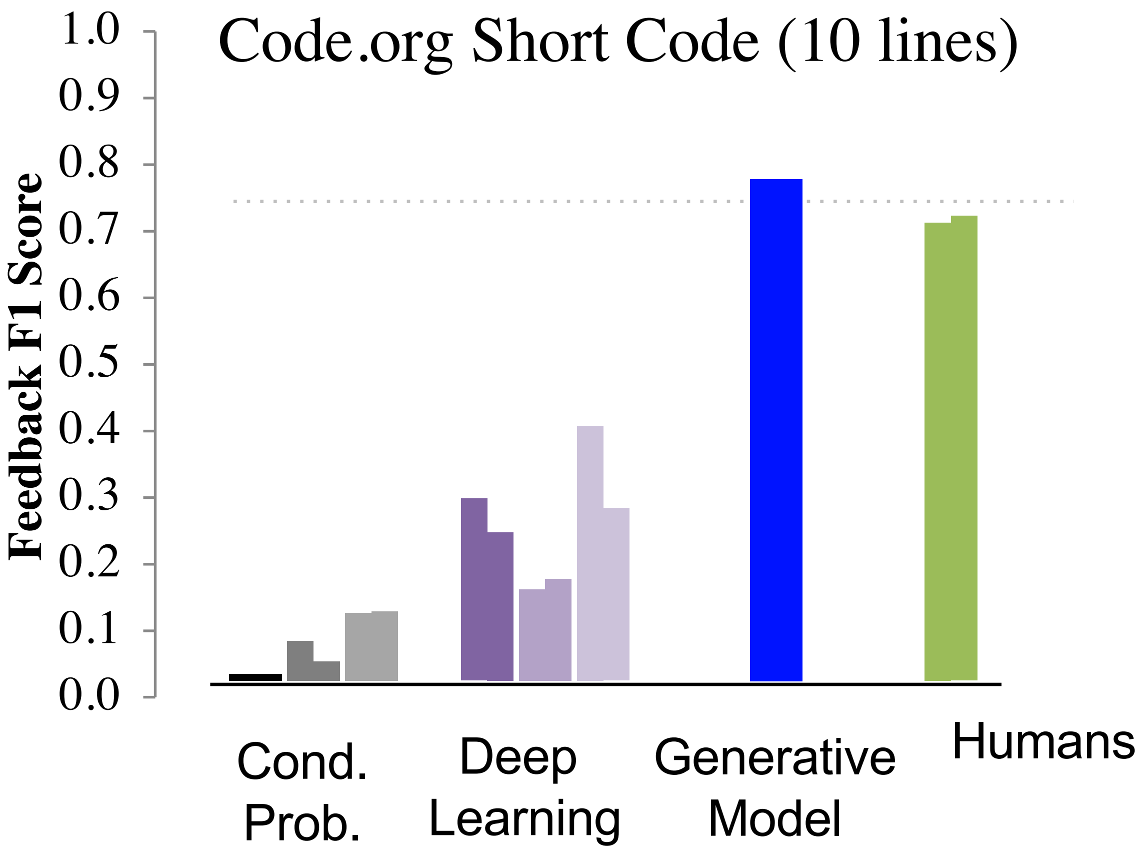


Generative Model of Characters

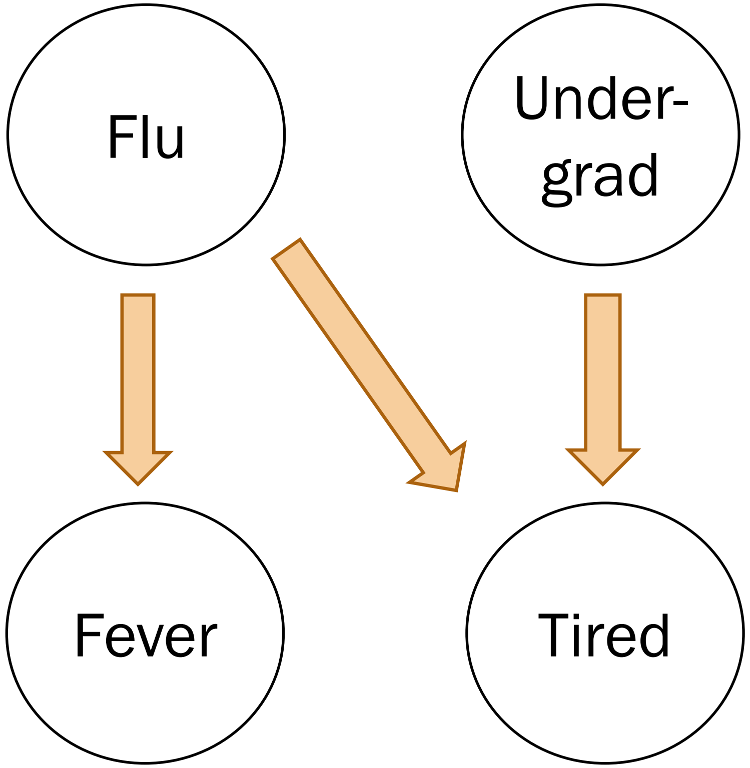
Lake et al, 2015



Computers Couldn't Understand Code



Constructing a Bayesian Network



$P(T = 1|F_{lu} = 0, U = 0)$
$P(T = 1|F_{lu} = 0, U = 1)$
$P(T = 1|F_{lu} = 1, U = 0)$
$P(T = 1|F_{lu} = 1, U = 1)$

In a Bayesian Network,

Each random variable is caused by its parents. Def P(node | parents)

- Node: random variable
- Directed edge: causality

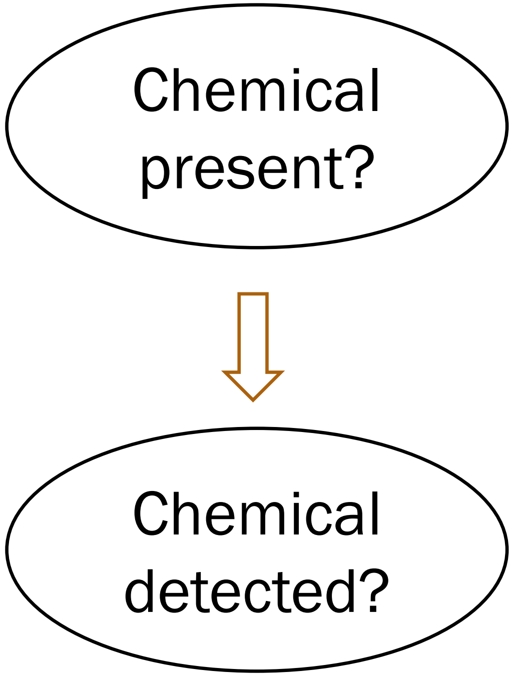
Examples:

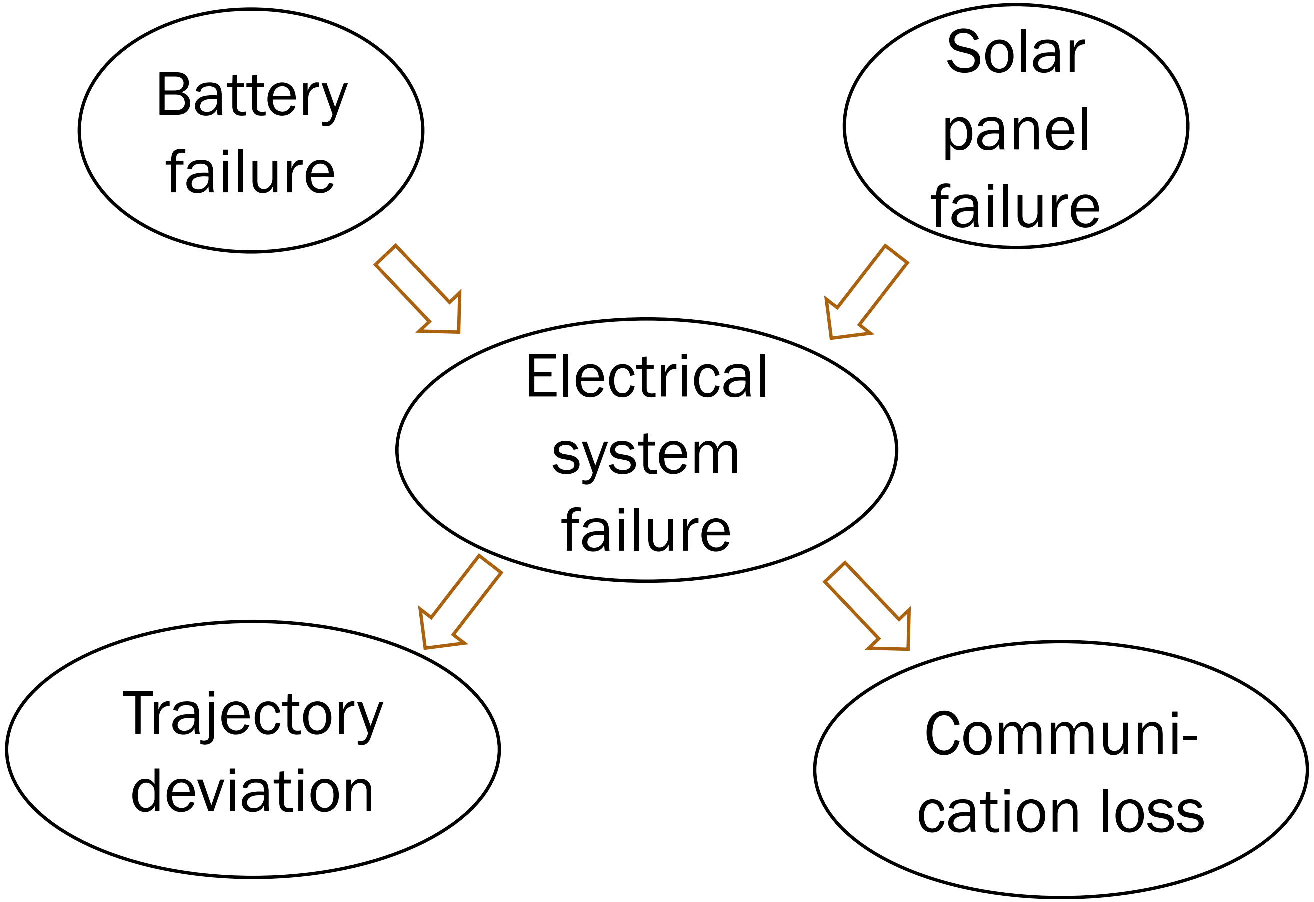
- $P(F_{lu} = 1)$
- $P(U = 0)$
- $P(F_{ev} = 1|F_{lu} = 1)$, $P(F_{ev} = 1|F_{lu} = 0)$
- $P(T = 1|F_{lu} = 0, U = 0)$ …

Make a *Generative* Model

A good probabilistic model is generative. It explains the process through which the joint is created.

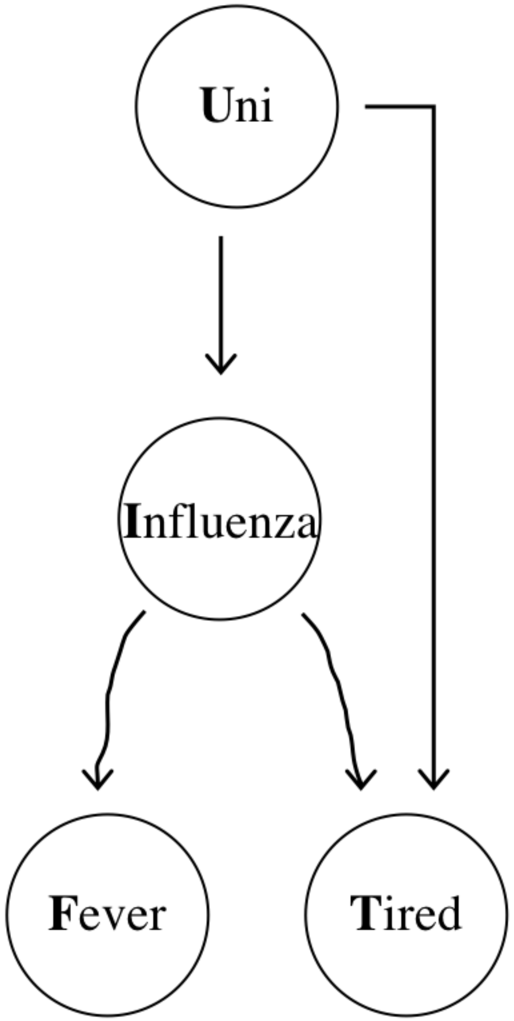
Other applications





Bayesian Network

Simple Disease Model




```
def get_prob_Xi(x, parents):
    # what is the probability that Xi = x
    # given the list parents of assignments to
    # the parents variables Xi
```

$P(\text{Uni} = 1) = 0.8$

$P(\text{Influenza} = 1 | \text{Uni} = 1) = 0.2$

$P(\text{Influenza} = 1 | \text{Uni} = 0) = 0.1$

$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 0) = 0.1$

$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 0) = 0.8$

$P(\text{Fever} = 1 | \text{Influenza} = 1) = 0.9$

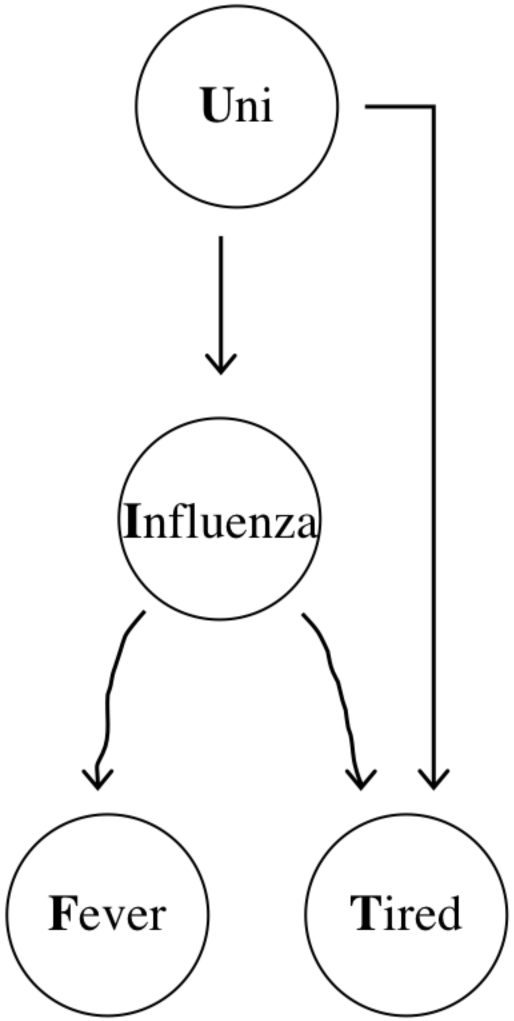
$P(\text{Fever} = 1 | \text{Influenza} = 0) = 0.05$

$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 1) = 0.9$

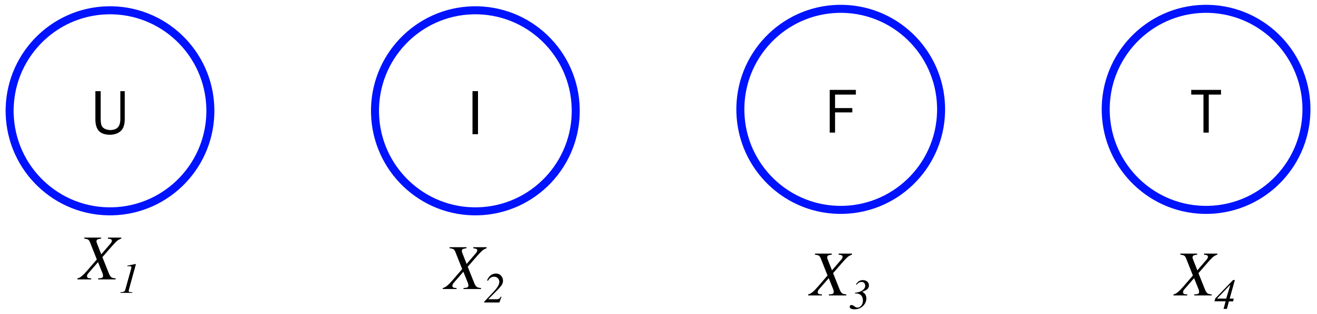
$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 1) = 1.0$

Bayesian Network Assumption

Simple Disease Model



Order nodes by ancestry



$$P(\text{Joint}) = \prod_i P(x_i | x_{i-1}, \ldots, x_1)$$

$$= \prod_i P(x_i | \text{Values of parents of } X_i)$$

Assume: Once you know the value of the parents of a variable in your network, X_i, any further information about non-descendents will not change your belief in X_i.

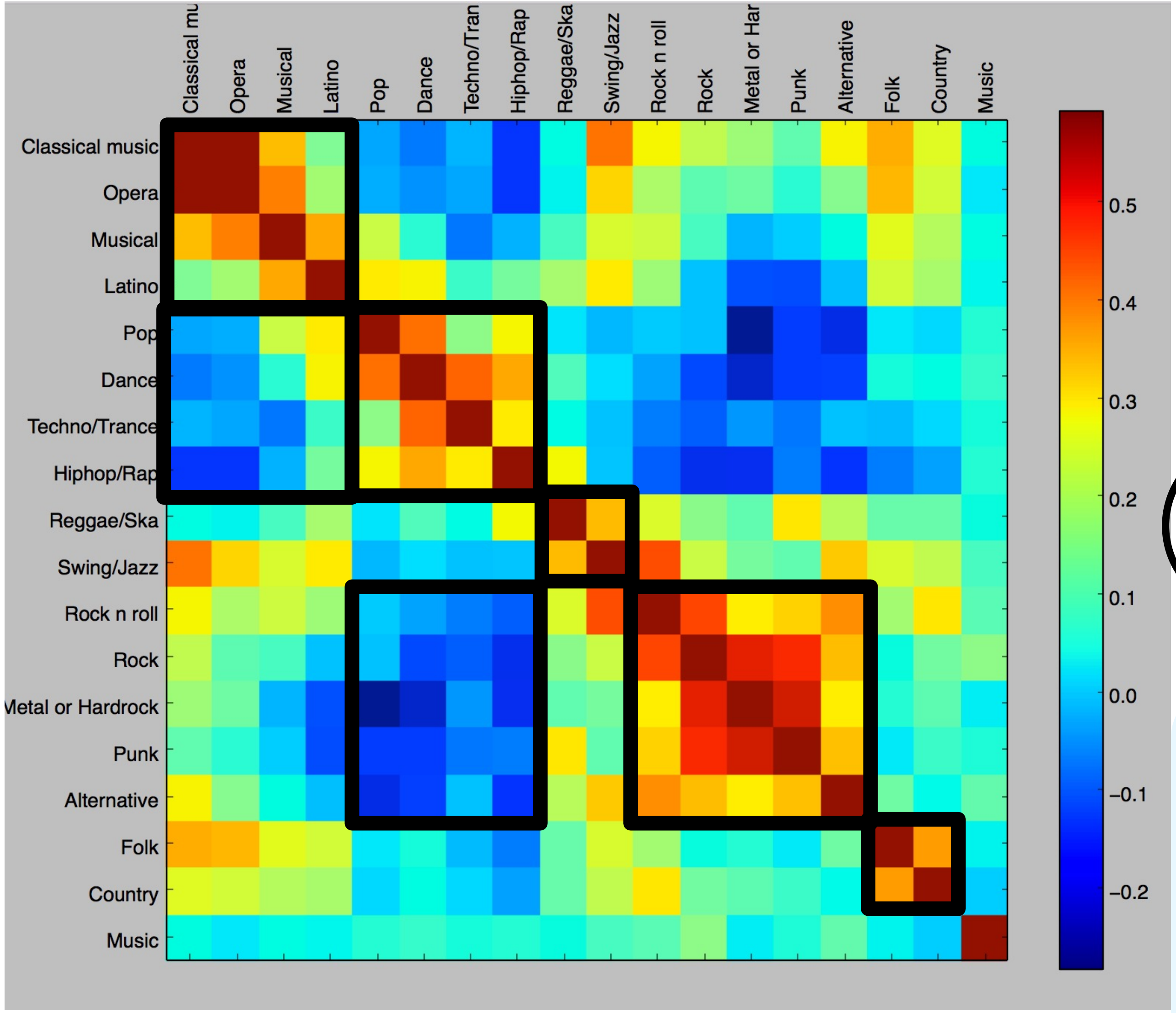
How do people design these?

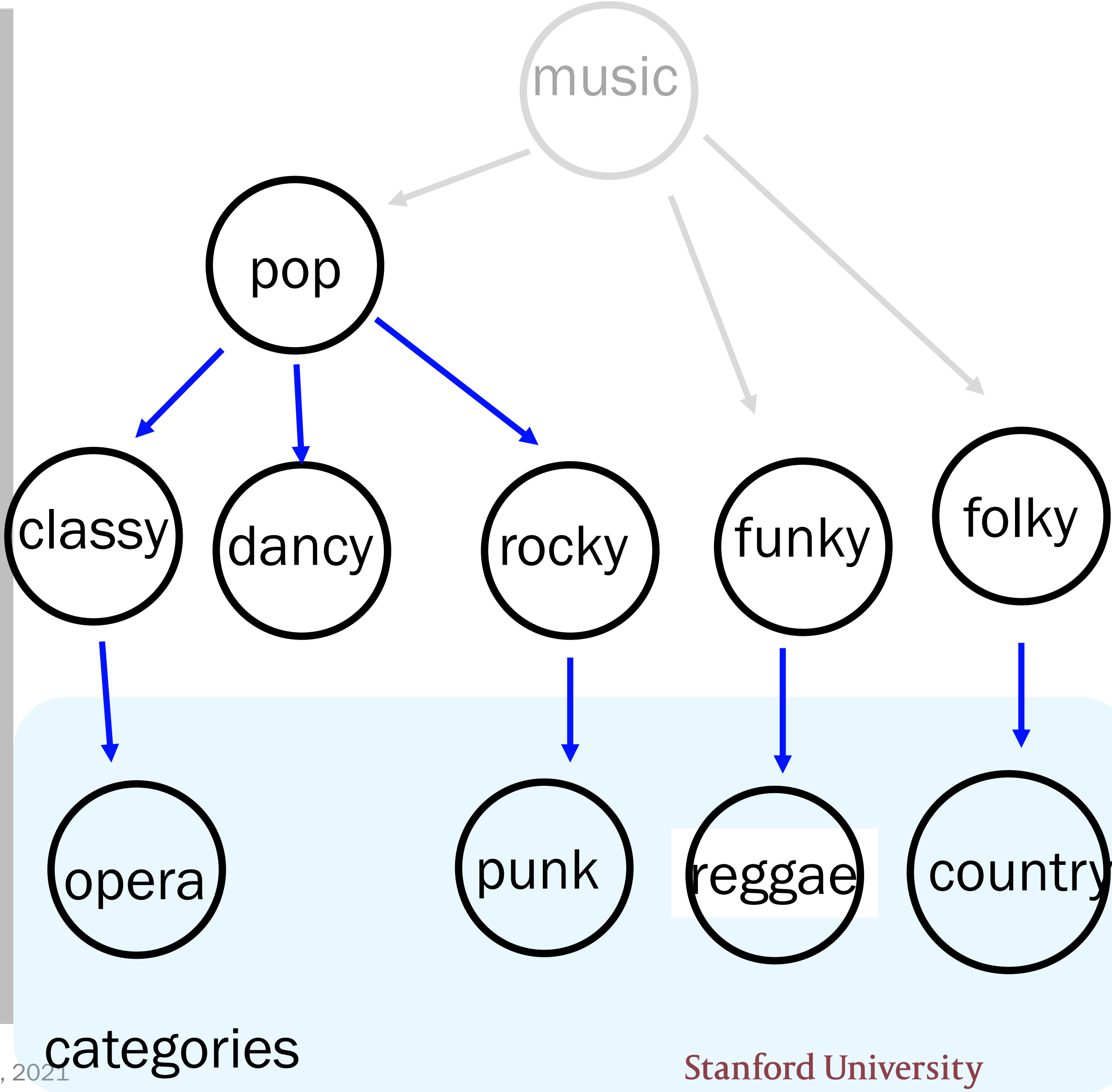
C15 | 3

	A	B	C	D	E	F	G	H
1	Music	Dance	Folk	Country	Classical music	Musical	Pop	Rock
2	5	2	1	2	2	1	5	5
3	4	2	1	1	1	2	3	5
4	5	2	2	3	4	5	3	5
5	5	2	1	1	1	1	2	2
6	5	4	3	2	4	3	5	3
7	5	2	3	2	3	3	2	5
8	5	5	3	1	2	2	5	3
9	5	3	2	1	2	2	4	5
10	5	3	1	1	2	4	3	5
11	5	2	5	2	2	5	3	5
12	5	3	2	1	2	3	4	3
13	5	1	1	1	4	1	2	5
14	5	1	2	1	4	3	3	5
15	5	5	3	2	1	5	5	2
16	5	2	1	1	2	3	4	5
17	1	2	2	3	4	3	3	5
18	5	3	1	1	1	2	4	4
19	5	3	3	3	2	2	4	4
20	5	5	4	3	4	5	5	4
21	5	3	3	2	4	2	2	4
22	5	3	2	3	4	3	2	5
23	5	1	1	3	2	2	2	5
24	5	3	2	3	3	3	4	
25	5	4	2	2	2	4	4	5
26	5	3	1	1	4	3	3	5
27	5	4	2	1	2	3	5	1
28	5	5	5	4	5	3	4	4
29	4	3	4	1	3	2	2	4
30	5	5	1	1	1	1	3	4
31	5	3	4	2	3	3	3	4
32	4	4	3	3	3	3	4	4
33	4	4	1	3	2	3	5	3
34	5	3	1	3	2	3	3	4
35	5	2	2	3	4	5	4	3

music

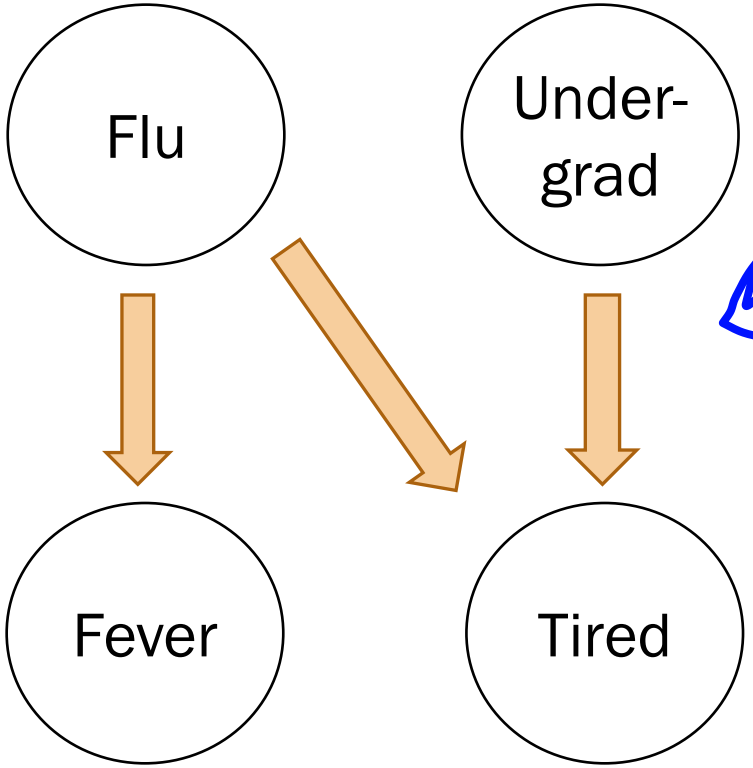
From Correlation to Bayes Net. Alternative!





The art of modelling

$P(F_{lu} = 1) = 0.1$ $\quad$ $P(U = 1) = 0.8$



1. Design this

2. Also design this. Later in CS109: learn this from data

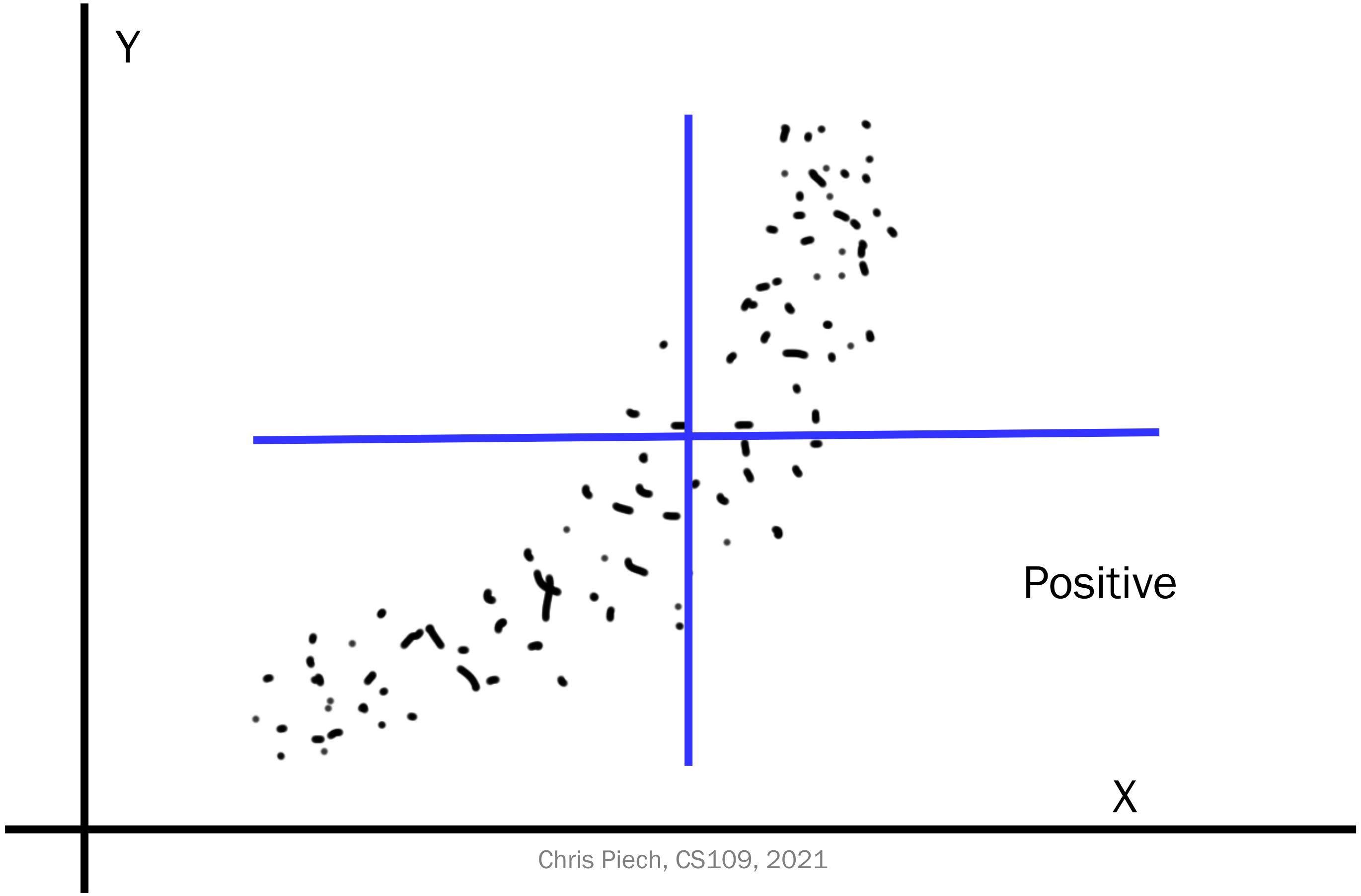
$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$
$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$
$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

Calculate the Covariance / Correlation (new stat!)

$$\text{Cov}(X,Y) = E[(X - E[X])(Y - E[Y])]$$

$$\text{Cov}(X,Y) = E[XY] - E[Y]E[X]$$



Correlation is just normalized Covariance

Correlation   Covariance

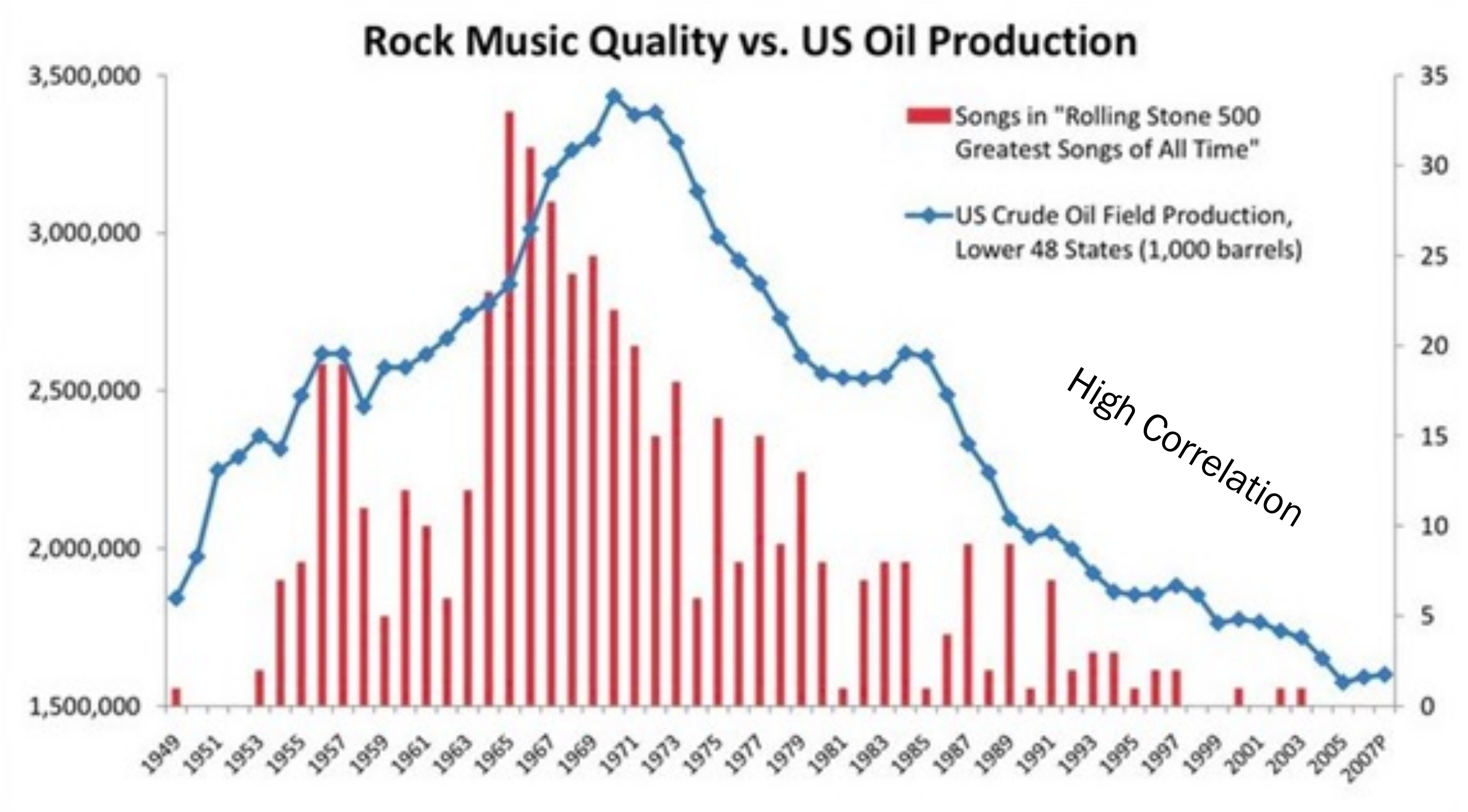
$$\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

It is always true that

$$\text{Cov}(X,Y) < \sqrt{\text{Var}(X)\text{Var}(Y)}$$

$$\text{Cov}(X,Y) > -\sqrt{\text{Var}(X)\text{Var}(Y)}$$

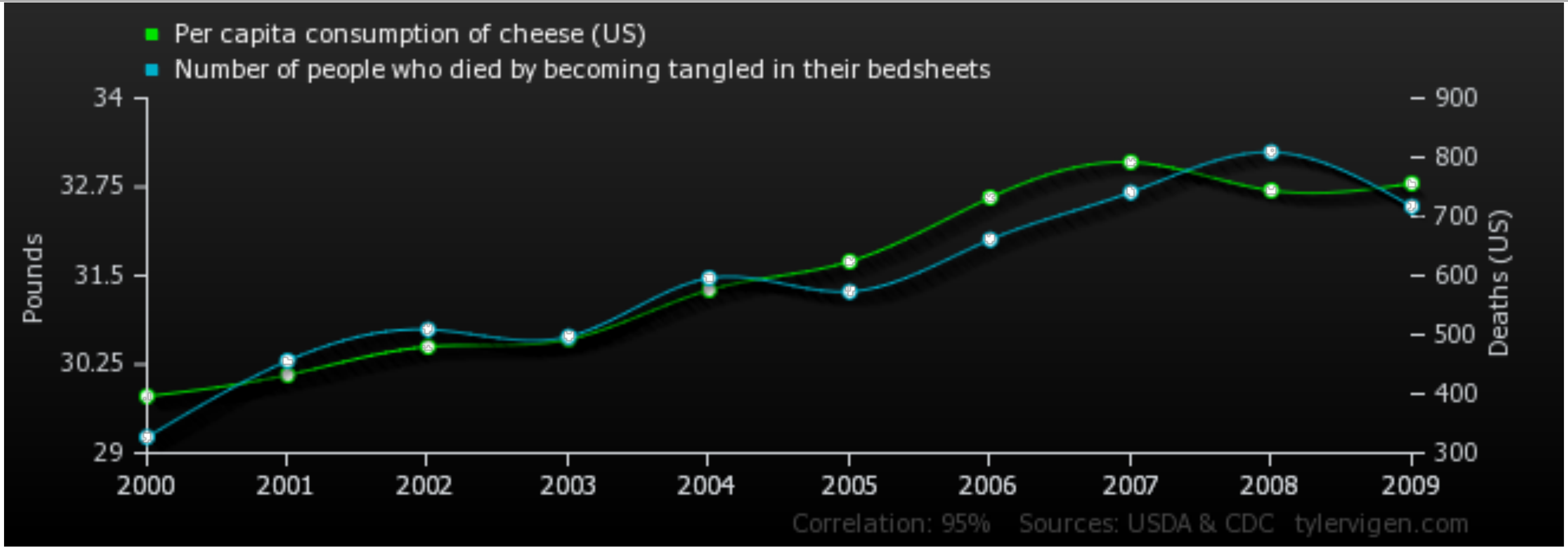
Rock Music Vs Oil?



Hubbert Peak Theory

http://www.aei.org/publication/blog/

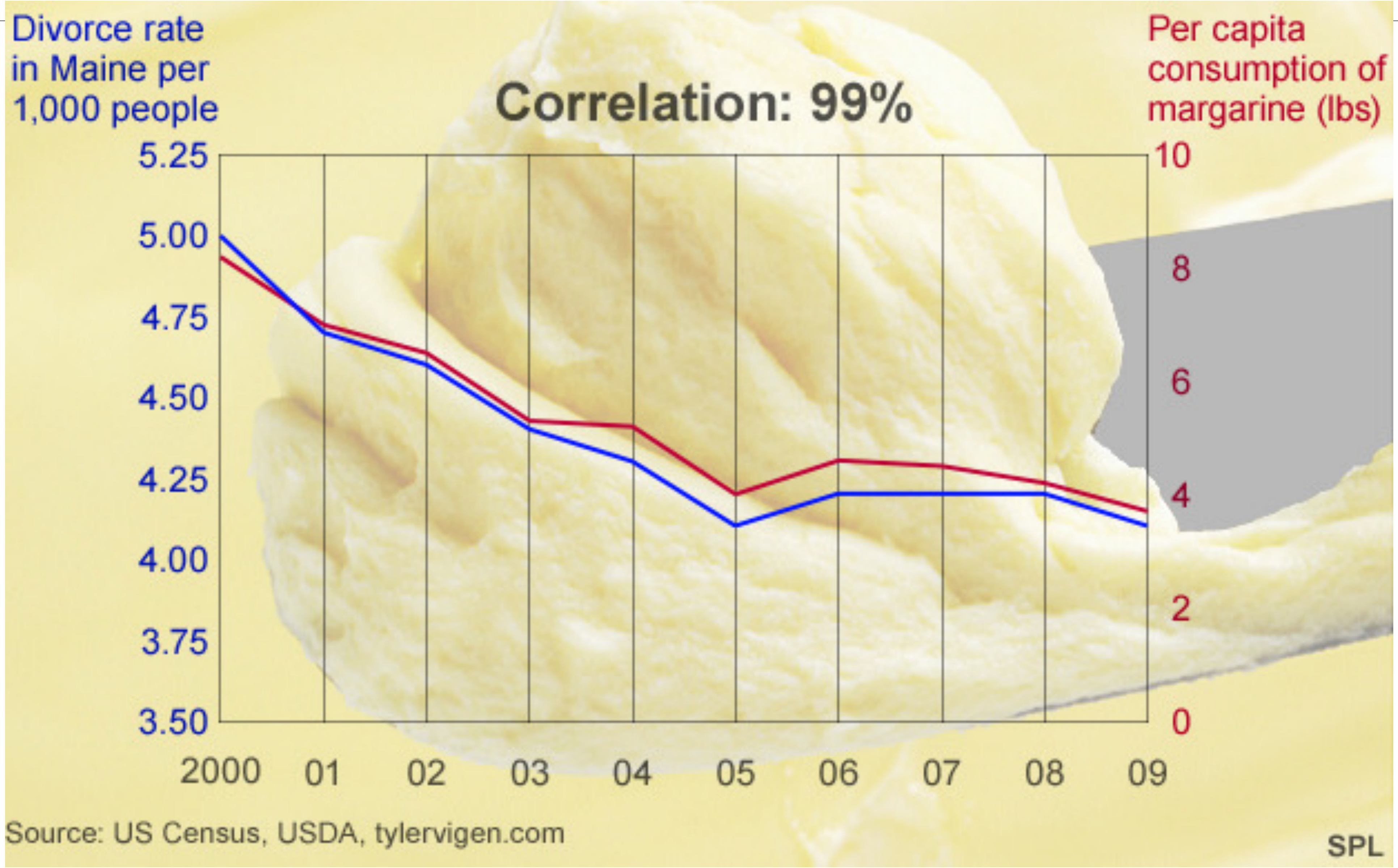
Tell your friends!



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Per capita consumption of cheese (US) *Pounds (USDA)*	29.8	30.1	30.5	30.6	31.3	31.7	32.6	33.1	32.7	32.8
Number of people who died by becoming tangled in their bedsheets *Deaths (US) (CDC)*	327	456	509	497	596	573	661	741	809	717

Correlation: 0.947091

Divorce Vs Butter?



http://www.bbc.com/news/magazine-27537142

Covariance of Zero Does Not Mean Independence!

X and Y are random variables with PMF:

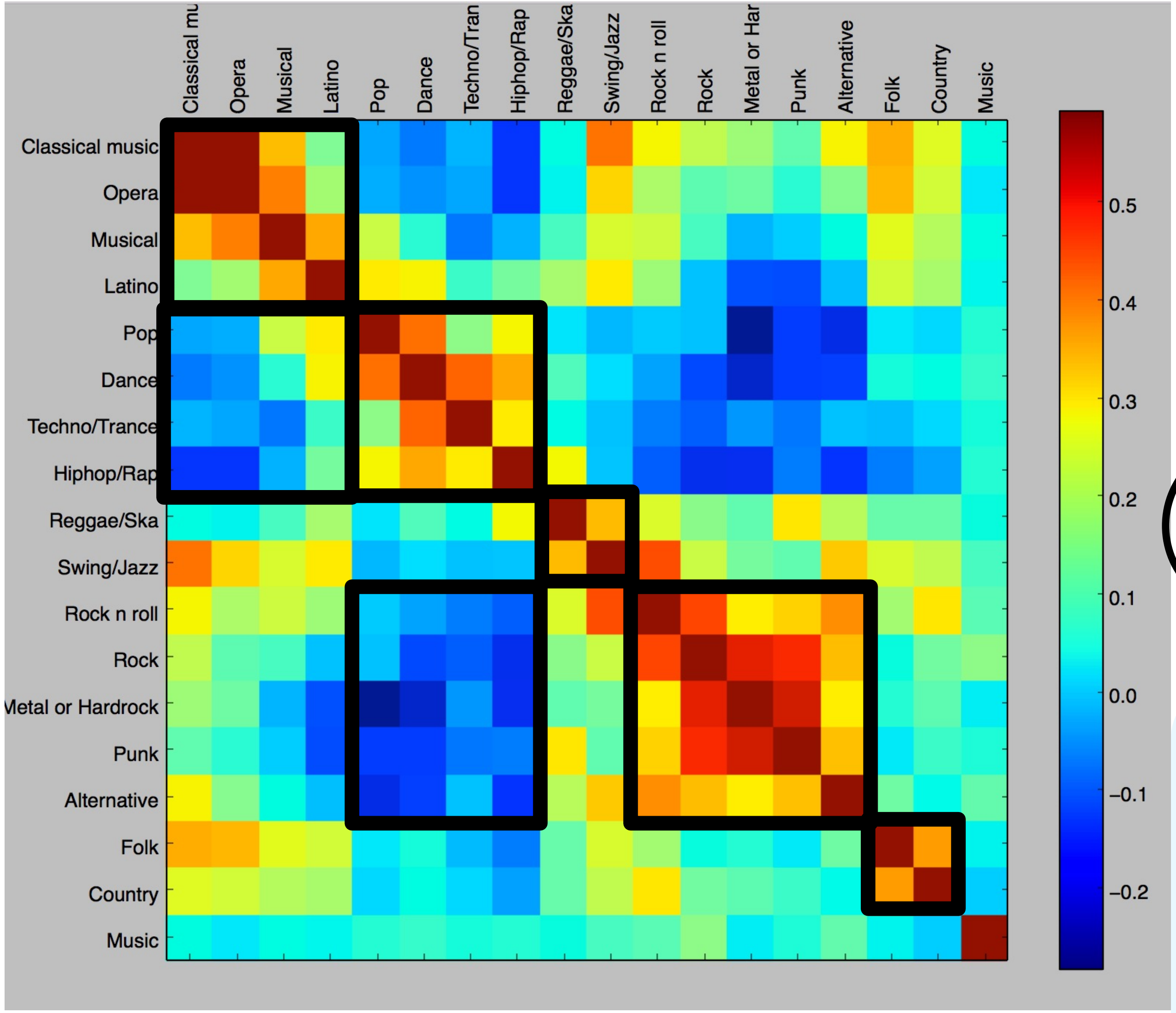
Y \ X	-1	0	1	$p_Y(y)$
0	1/3	0	1/3	2/3
1	0	1/3	0	1/3
$p_X(x)$	1/3	1/3	1/3	1

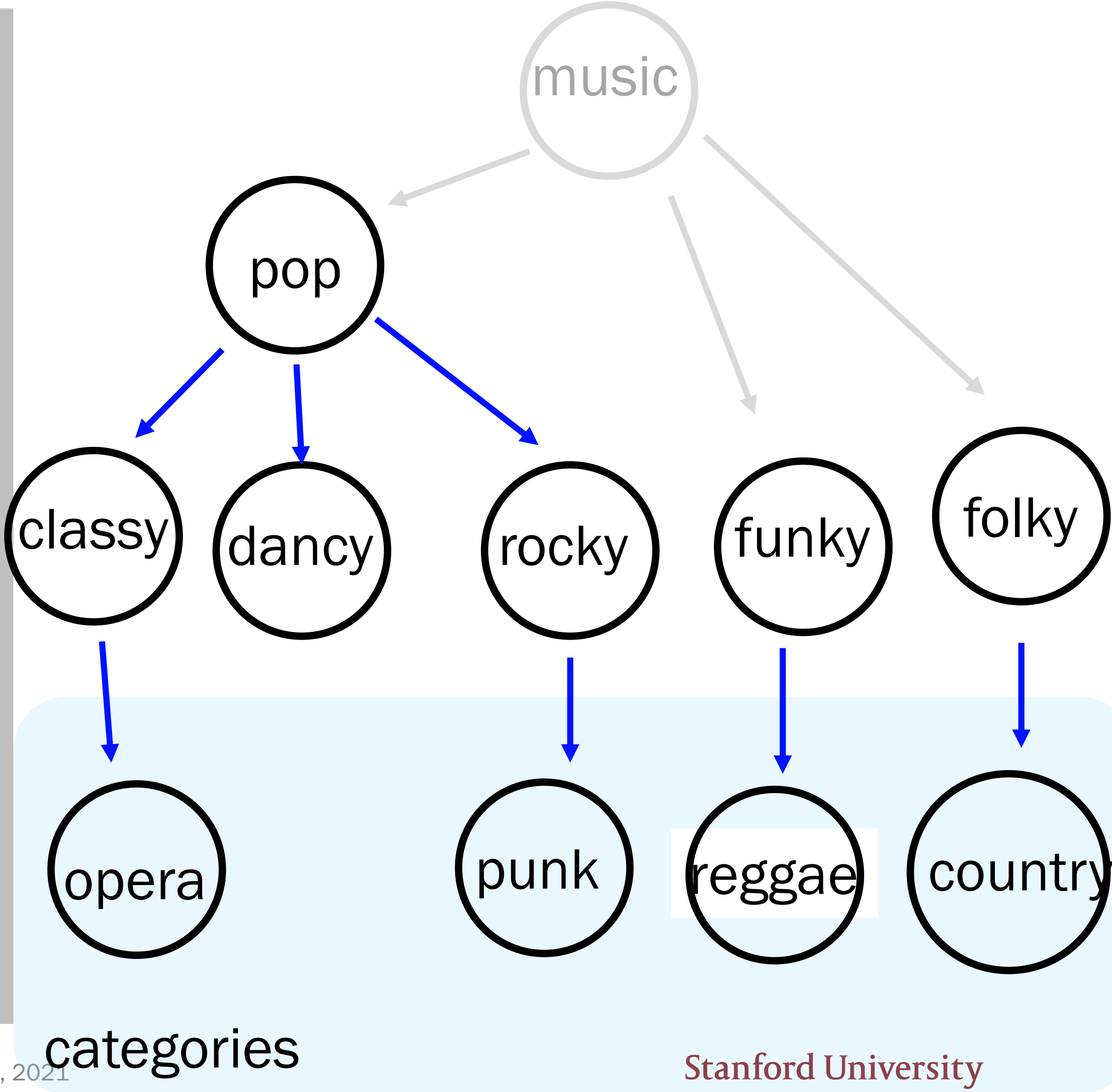
$$Y = \begin{cases} 0 & \text{if } X \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

- E[X] = -1(1/3) + 0(1/3) + 1(1/3) = 0
- E[Y] = 0(2/3) + 1(1/3) = 1/3
- Since XY = 0, E[XY] = 0
- Cov(X, Y) = E[XY] – E[X]E[Y] = 0 – 0 = 0

But, X and Y are clearly dependent!

Recall: It is a useful starting point





We have models. Need to solve problems

Inference

Monday: Inference

Inference *noun*

An updated belief about a random variable (or multiple) based on conditional knowledge regarding another random variable (or multiple) in a probabilistic model.

TLDR: conditional probability with random variables.

All the Bayes Belong to Us

M,N are discrete. X, Y are continuous

OG Bayes

$$p_{M|N}(m|n) = \frac{P_{N|M}(n|m)p_M(m)}{p_N(n)}$$

Mix Bayes #1

$$f_{X|N}(x|n) = \frac{P_{N|X}(n|x)f_X(x)}{P_N(n)}$$

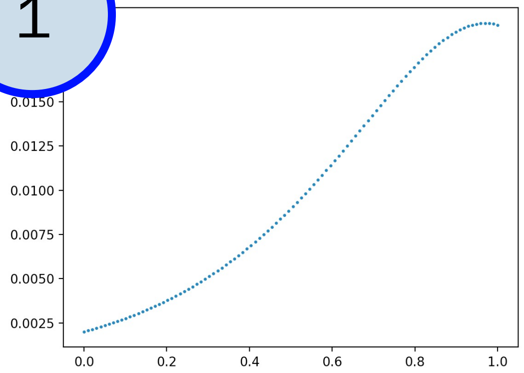
Mix Bayes #2

$$p_{N|X}(n|x) = \frac{f_{X|N}(x|n)p_N(n)}{f_X(x)}$$

$$f_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)f_X(x)}{f_Y(y)}$$

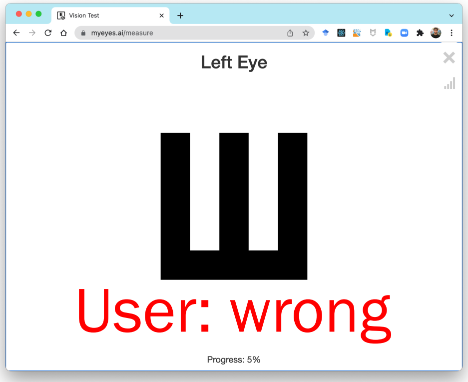
Inference on a non-bernoulli random variable

1



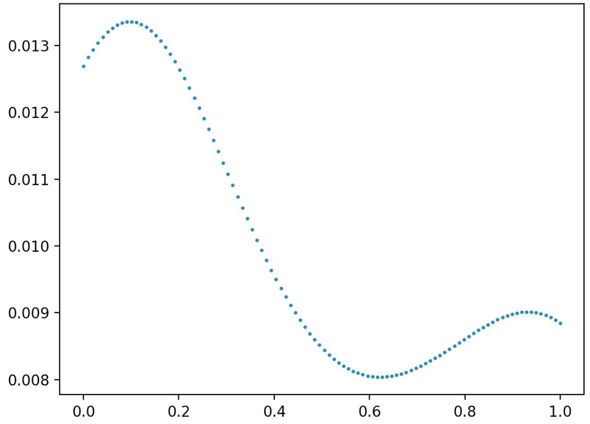
$$P(A = a)$$

2



Observation $Y = 0$

3



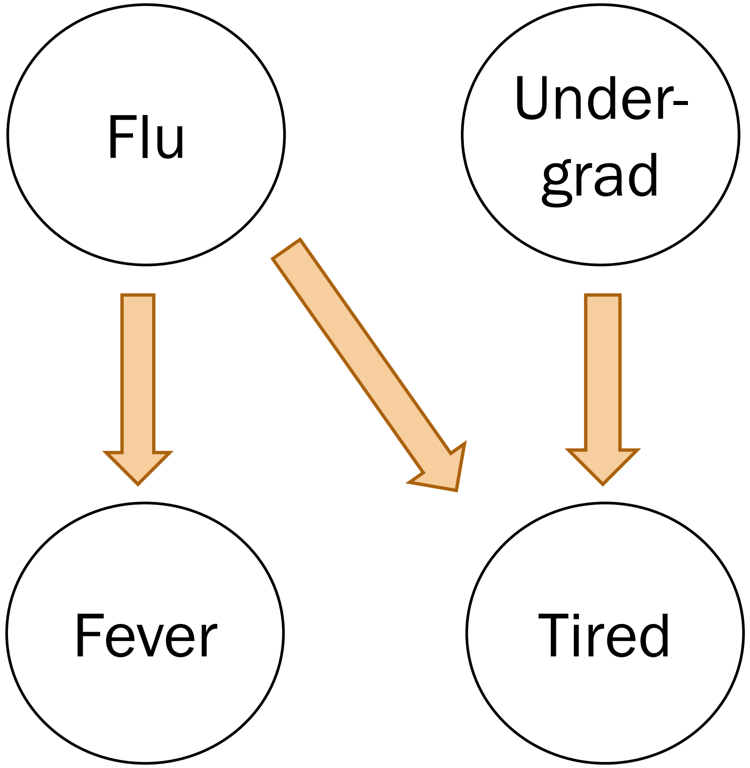
$$P(A = a | Y = 0)$$

We can perform **inference** when there are two random variables using Bayes!

End Review

Inference: Algebra

Bayes Nets: Conditional independence



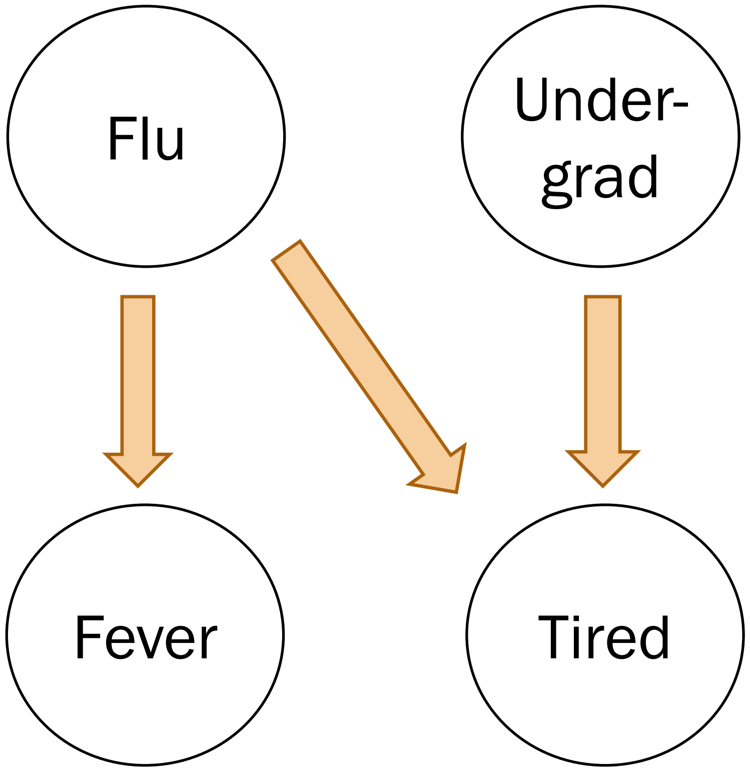
In a Bayesian Network,

Each random variable is conditionally independent of its non-descendants, given its parents.

- Node: random variable
- Directed edge: conditional dependency

Inference via math

$P(F_{lu} = 1) = 0.1$ $\quad$ $P(U = 1) = 0.8$



$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

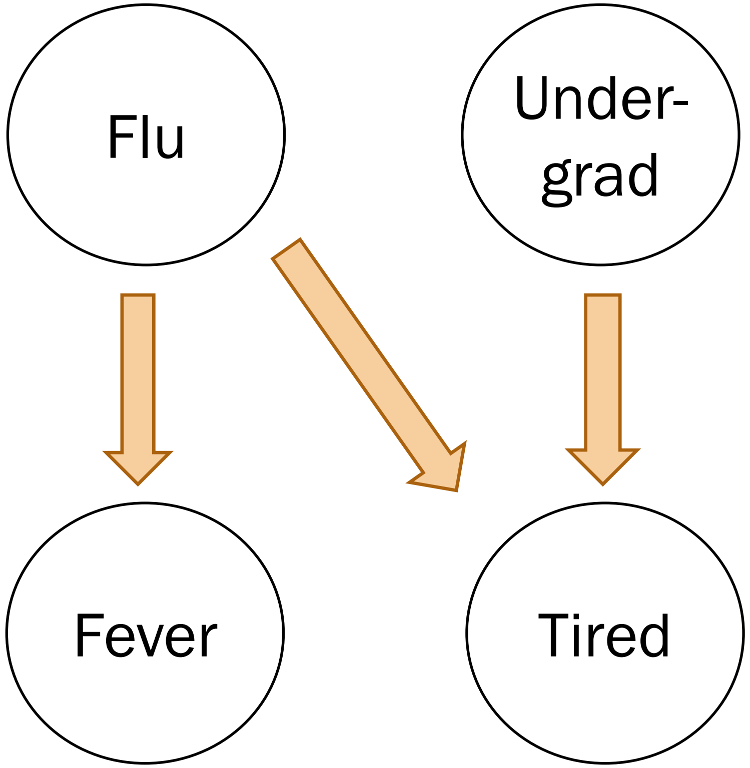
$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

1. $P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)$?

   Compute joint probabilities using chain rule.

Inference via math

$P(F_{lu} = 1) = 0.1$ $P(U = 1) = 0.8$



$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$
$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$
$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

2. $P(F_{lu} = 1 | F_{ev} = 0, U = 0, T = 1)$?

   1. Compute joint probabilities

      $P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)$

      $P(F_{lu} = 0, F_{ev} = 0, U = 0, T = 1)$

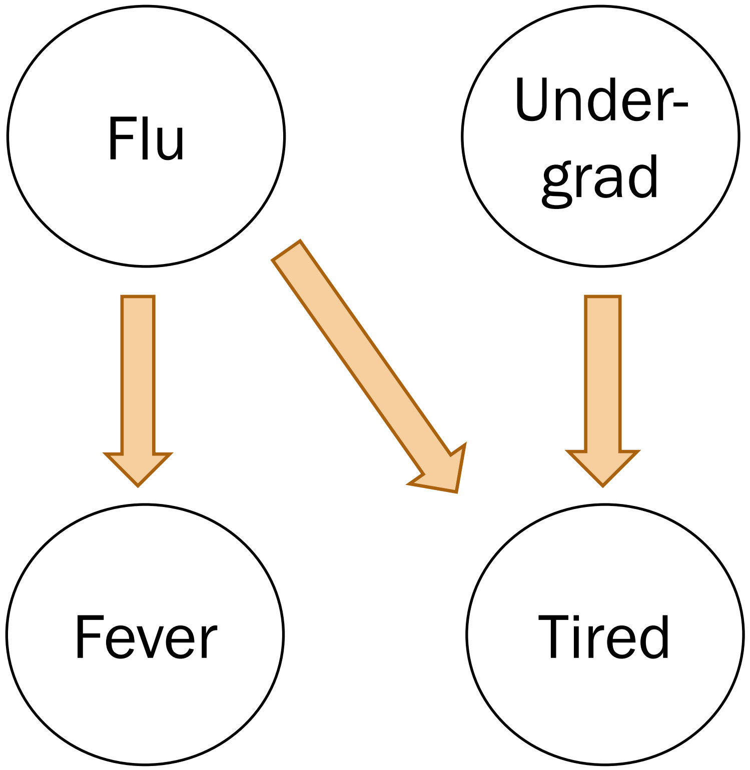
   2. Definition of conditional probability

$$\frac{P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)}{\sum_x P(F_{lu} = x, F_{ev} = 0, U = 0, T = 1)}$$

$$= 0.095$$

Inference via math

$P(F_{lu} = 1) = 0.1$ $\quad$ $P(U = 1) = 0.8$



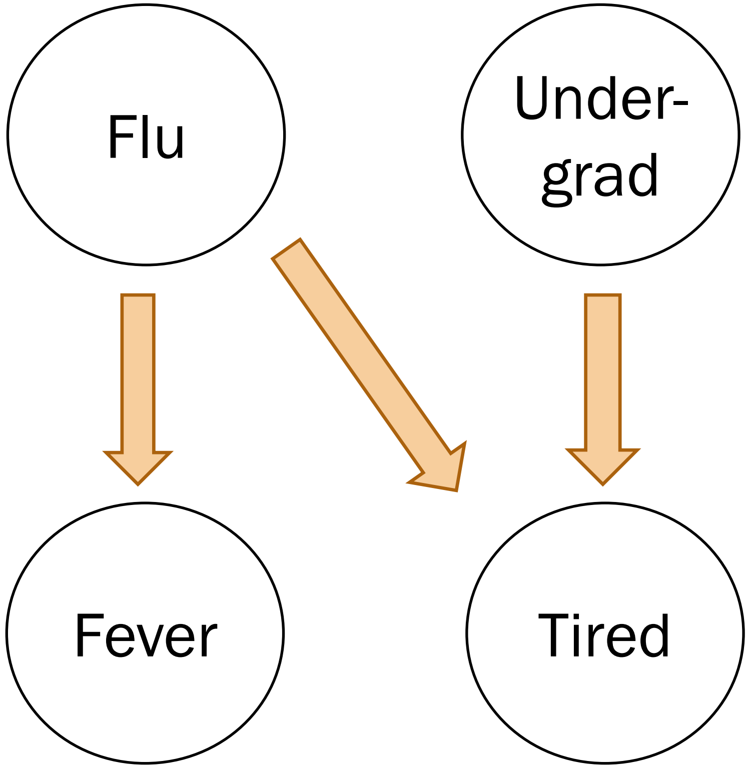
$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$
$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$
$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

3. $P(F_{lu} = 1 | U = 1, T = 1)$?

Inference via math

$P(F_{lu} = 1) = 0.1$ $\qquad$ $P(U = 1) = 0.8$



$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$
$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$
$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

3. $P(F_{lu} = 1 | U = 1, T = 1)$?

   1. Compute joint probabilities

      $$P(F_{lu} = 1, U = 1, F_{ev} = 1, T = 1)$$

      ...

      $$P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1)?$$

   2. Definition of conditional probability

      $$\frac{\sum_y P(F_{lu} = 1, U = 1, F_{ev} = y, T = 1)}{\sum_x \sum_y P(F_{lu} = x, U = 1, F_{ev} = y, T = 1)}$$

$$= 0.122$$

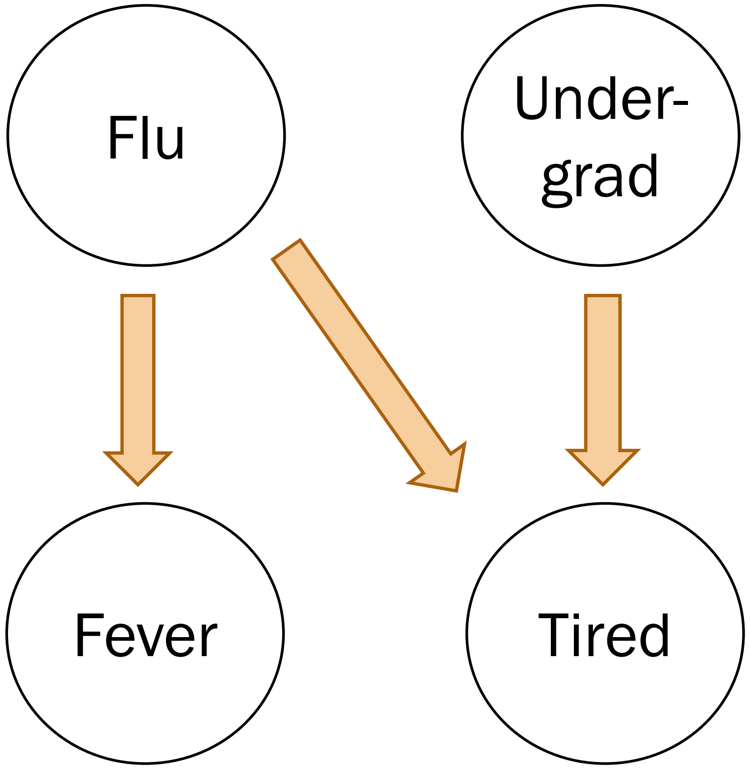
Rejection sampling algorithm

Step 0:

Have a fully specified Bayesian Network

$P(F_{lu} = 1) = 0.1$

$P(U = 1) = 0.8$



$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

$P(T = 1 | F_{lu} = 0, U = 0) = 0.1$
$P(T = 1 | F_{lu} = 0, U = 1) = 0.8$
$P(T = 1 | F_{lu} = 1, U = 0) = 0.9$
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

Alg #0: Straight Math

Too many possible inference questions one could ask…

Alg #1: Rejection Sampling

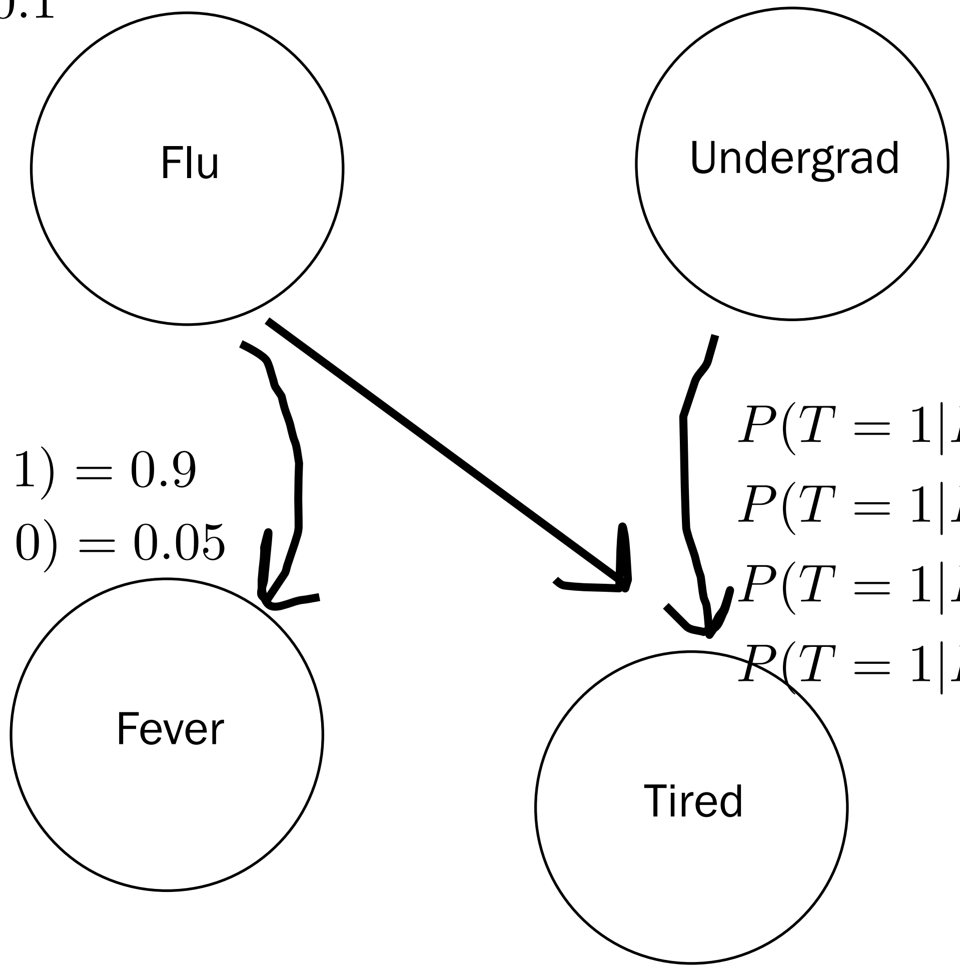
```
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any probability question
# with multivariate samples from the joint,
# where conditioned variables match
def main():
    obs = getObservation()
    print 'Observation = ', obs

    samples = sampleATon()
    prob = probFluGivenObs(samples, obs)
    print 'Pr(Flu) = ', prob
```

```python
# Method: Sample A Ton
# ---------------------
# chose N_SAMPLES with likelihood proportional
# to the joint distribution
def sampleATon():
    samples = []
    for i in range(N_SAMPLES):
        sample = makeSample()
        samples.append(sample)
    return samples
```

Recall: Probabilistic Model



$$P(Fl = 1) = 0.1$$

$$P(U = 1) = 0.8$$

$$P(Fev = 1|Flu = 1) = 0.9$$
$$P(Fev = 1|Flu = 0) = 0.05$$

$$P(T = 1|Flu = 0, U = 0) = 0.1$$
$$P(T = 1|Flu = 0, U = 1) = 0.8$$
$$P(T = 1|Flu = 1, U = 0) = 0.9$$
$$P(T = 1|Flu = 1, U = 1) = 1.0$$

```python
# Method: Make Sample
# ------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

    # choose fever based on flue
    if flu == 1: fev = bern(0.9)
    else:        fev = bern(0.05)

    # choose tired based on (undergrade and flu)
    if und == 1 and flu == 1:   tir = bern(1.0)
    elif und == 1 and flu == 0: tir = bern(0.8)
    elif und == 0 and flu == 1: tir = bern(0.9)
    else:                       tir = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

```
# Method: Make Sample
# ------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

    # choose fever based on flue
    if flu == 1: fev = bern(0.9)
    else:        fev = bern(0.05)

    # choose tired based on (undergrade and flu)
    if und == 1 and flu == 1:   tir = bern(1.0)
    elif und == 1 and flu == 0: tir = bern(0.8)
    elif und == 0 and flu == 1: tir = bern(0.9)
    else:                       tir = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

```python
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical 
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

    # choose fever based on flue
    if flu == 1: fev = bern(0.9)
    else:        fev = bern(0.05)

    # choose tired based on (undergrade and flu)
    if und == 1 and flu == 1:   tir = bern(1.0)
    elif und == 1 and flu == 0: tir = bern(0.8)
    elif und == 0 and flu == 1: tir = bern(0.9)
    else:                       tir = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

```python
# Method: Make Sample
# -------------------
# chose a single sample from the joint distribut
# based on the medical "Probabilistic Graphical
def makeSample():
    # prior on causal factors
    flu = bern(0.1)
    und = bern(0.8)

    # choose fever based on flue
    if flu == 1: fev = bern(0.9)
    else:        fev = bern(0.05)

    # choose tired based on (undergrade and flu)
    if und == 1 and flu == 1:   tir = bern(1.0)
    elif und == 1 and flu == 0: tir = bern(0.8)
    elif und == 0 and flu == 1: tir = bern(0.9)
    else:                       tir = bern(0.1)

    # a sample from the joint has an
    # assignment to *all* random variables
    return [flu, und, fev, tir]
```

Alg #1: Rejection Sampling

```
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any pro
# with multivariate sam
# where conditioned var
def main():
    obs = getObservatio
    print 'Observation 

    samples = sampleATo
    prob = probFluGiven
    print 'Pr(Flu) = ',
```

webMd — -bash — 30×20

```
[0, 1, 0, 1]
[1, 1, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[1, 1, 0, 1]
```

Alg #1: Rejection Sampling

```python
N_SAMPLES = 100000

# Program: Joint Sample
# ---------------------
# we can answer any probability question
# with multivariate samples from the joint,
# where conditioned variables match
def main():
    obs = getObservation()
    print 'Observation = ', obs

    samples = sampleATon()
    prob = probFluGivenObs(samples, obs)
    print 'Pr(Flu) = ', prob
```

```python
# Method: Probability of Flu Given Observation
# ---------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

```python
# Method: Probability of Flu Given Observation
# ---------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

```python
# Method: Probability of Flu Given Observation
# ---------------------------
# Calculate the probability of flu given many
# samples from the joint distribution and a set
# of ovservations to condition on.
def probFluGivenObs(samples, obs):
    # reject all samples which don't align
    # with condition
    keepSamples = []
    for sample in samples:
        if checkObsMatch(sample, obs):
            keepSamples.append(sample)

    # from remaining, simply count...
    fluCount = 0
    for sample in keepSamples:
        [flu, und, fev, tir] = sample
        if flu == 1:
            fluCount += 1

    # counting can be so sweet...
    return float(fluCount) / len(keepSamples)
```

Alg #1: Rejection Sampling

```
N_SAMPLES = 100000

# Program: Joint Sample
# ----------------------
# we can answer any pro
# with multivariate sam
# where conditioned var
def main():
    obs = getObservatio
    print 'Observation 

    samples = sampleATo
    prob = probFluGiven
    print 'Pr(Flu) = ',
```

webMd — -bash — 30×20

```
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 1]
[1, 1, 0, 1]
Pr(Flu) =  0.141503173687
> 
```

Lets try it!

BACK TO THE CODE

To the code!

Rejection sampling algorithm

Inference question: What is $P(F_{lu} = 1 | U = 1, T = 1)$?

$$\text{probability} \approx \frac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$$

Why would this definition of approximate probability make sense?

Why would this approximate probability make sense?

Inference question: What is $P(F_{lu} = 1 | U = 1, T = 1)$?

$$\text{probability} \approx \frac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$$

Recall our definition of probability as a frequency:

$$P(E) = \lim_{n \to \infty} \frac{n(E)}{n}$$

n = # of total trials

$n(E)$ = # trials where E occurs

```
webMd — -bash — 39×20
[0, 1, 1, 0]
[1, 0, 1, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 0]
[0, 1, 1, 0]
[1, 1, 1, 1]
[0, 1, 0, 0]
[0, 0, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 1, 0, 1]
[0, 1, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 1]
Observation =  [None, None, None, None]
Pr(Flu | Obs) =  0.10164
>
```

If you can sample enough from the joint distribution, you can answer any probability question

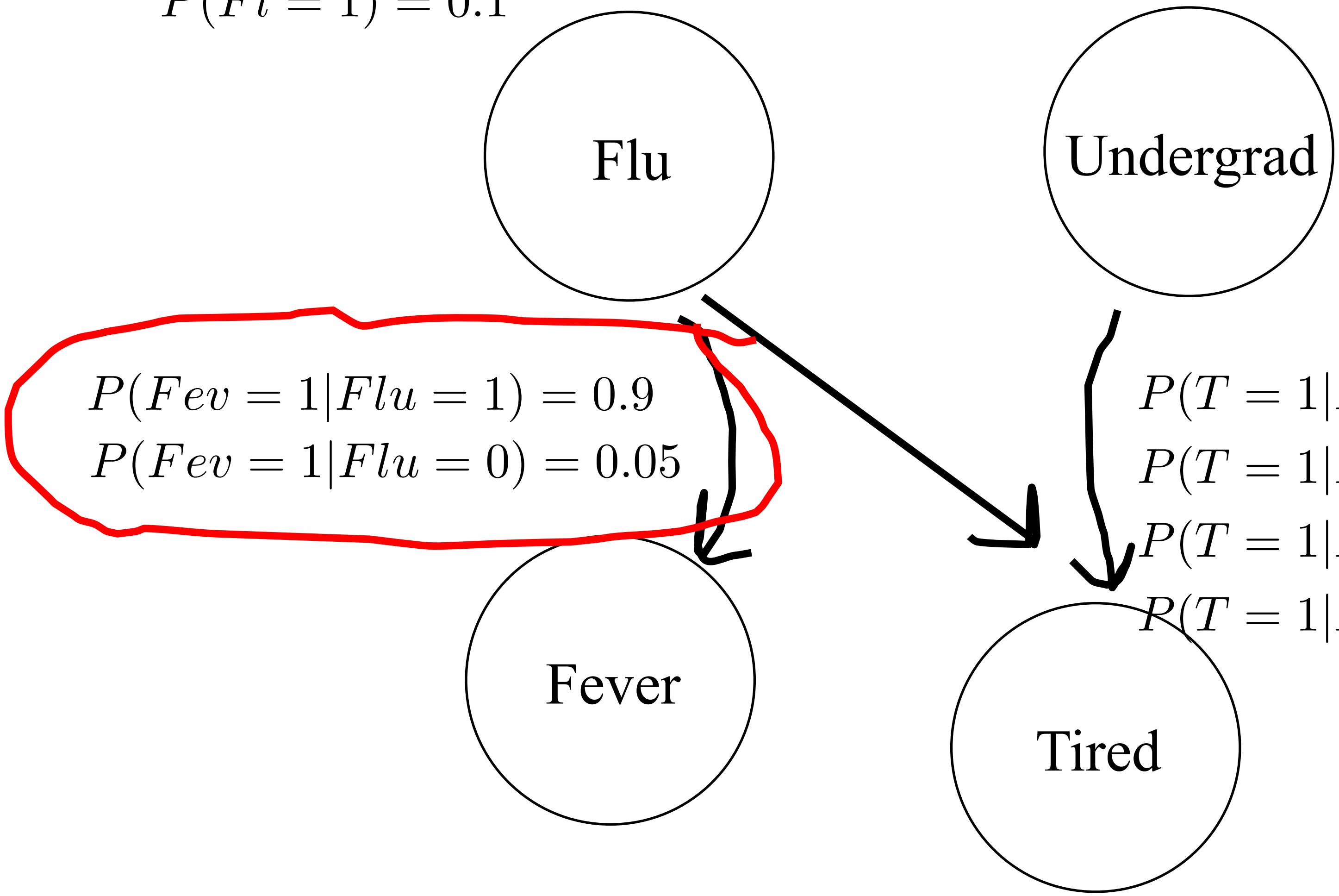
Each one of these is one joint sample:
[Flu, Undergrad, Fever, Tired]

What's the matter with joint sampling?

Probabilistic Model

$$P(Fl = 1) = 0.1$$

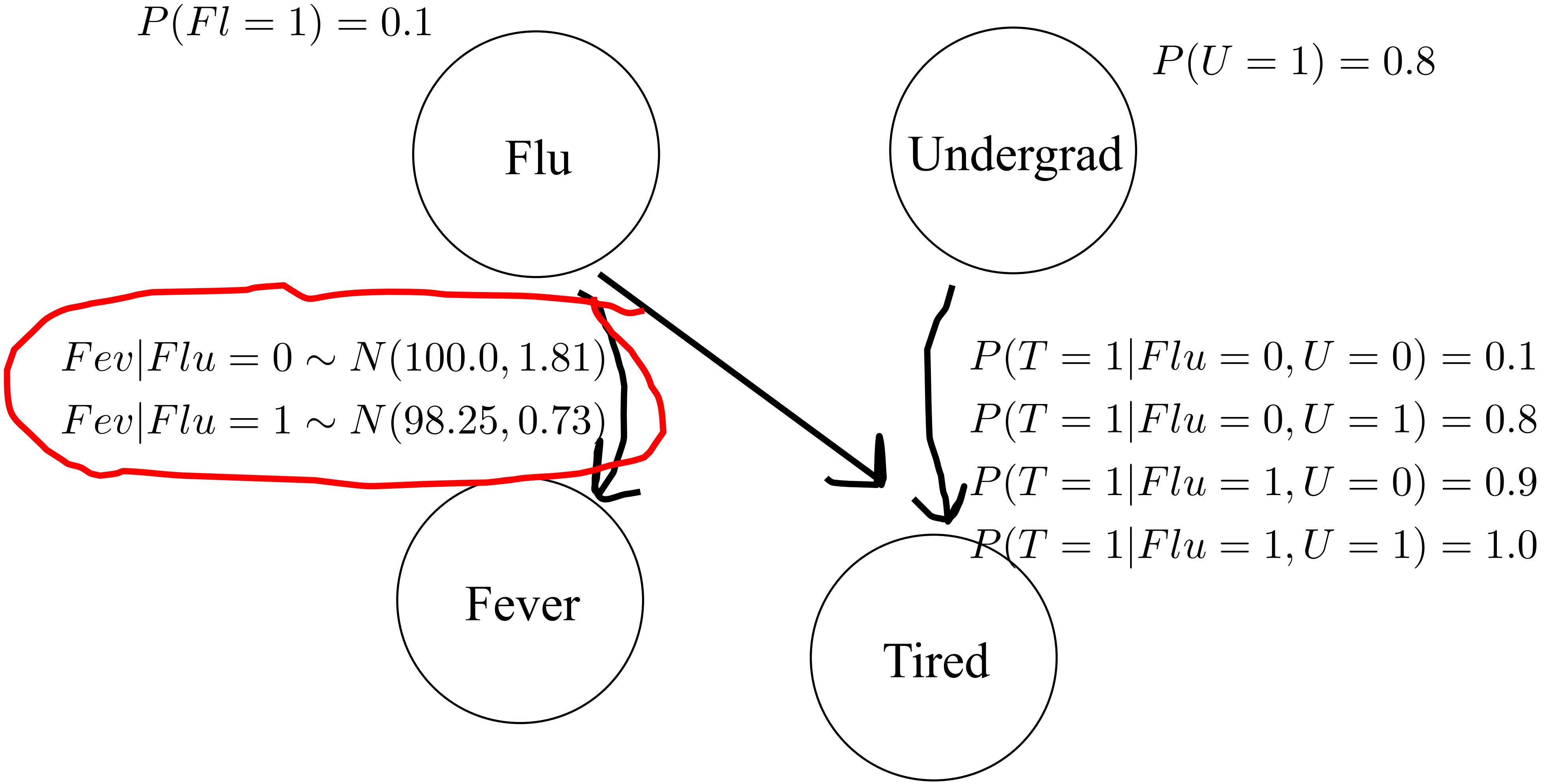
$$P(U = 1) = 0.8$$



$$P(Fev = 1 | Flu = 1) = 0.9$$
$$P(Fev = 1 | Flu = 0) = 0.05$$

$$P(T = 1 | Flu = 0, U = 0) = 0.1$$
$$P(T = 1 | Flu = 0, U = 1) = 0.8$$
$$P(T = 1 | Flu = 1, U = 0) = 0.9$$
$$P(T = 1 | Flu = 1, U = 1) = 1.0$$

Probabilistic Model




The Magic School Bus™
Blast Off!

Markov Chain

MCMC

Monte Carlo

Alg #2: MCMC

```
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[0, 1, 101.0, 0]
[0, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 0]
[1, 0, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
[1, 0, 101.0, 1]
[1, 1, 101.0, 1]
[1, 1, 101.0, 1]
Pr(Flu) = 0.9773
>
```

MCMC is a way to sample with conditioned variables fixed

Each one of these is one posterior sample:

[Flu, Undergrad, Fever, Tired]

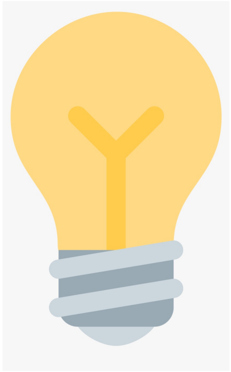
Many Algorithms

Rejection Sampling

MCMC

Pyro

Idea2Text



Stanford Acuity Test?

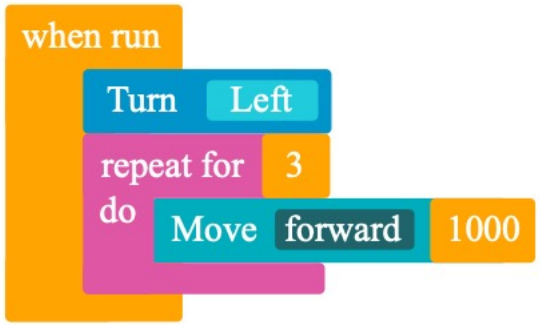
Version of rejection sampling

What about Code.org?

Computers Couldn’t Understand Code

60,000 students attempted this problem
37,000 unique solutions

Challenge

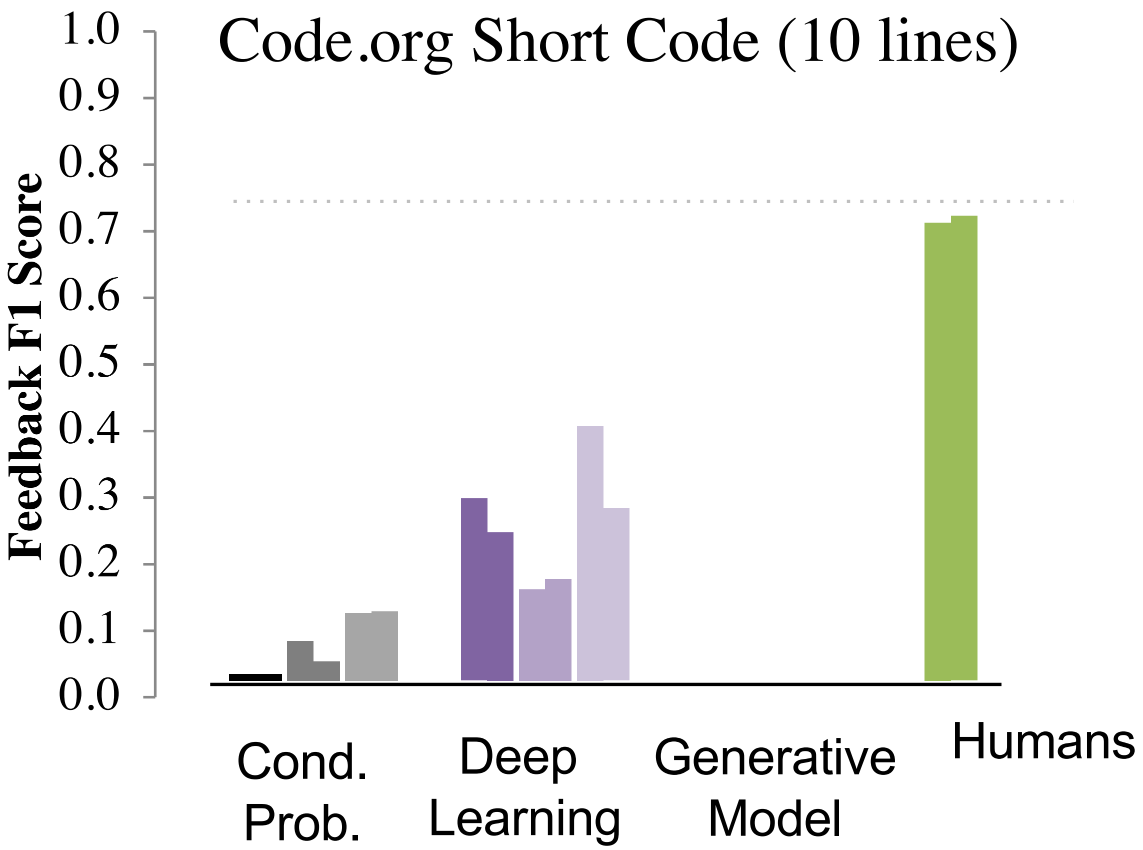


Student Code

You need to move and turn in your loop

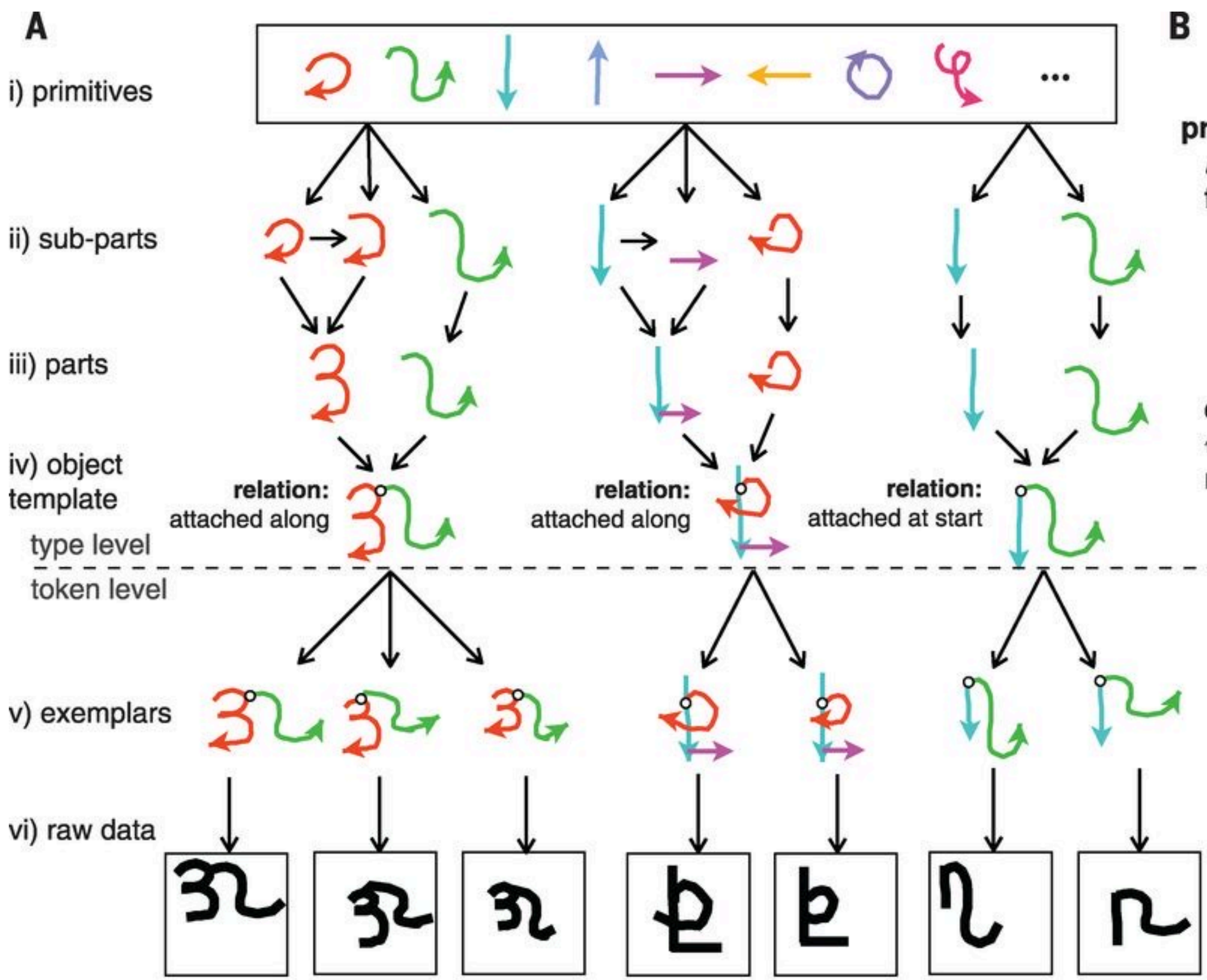
Insight

Computers Couldn't Understand Code



Generative Model of Handwritten Digits

Lake et al, 2015



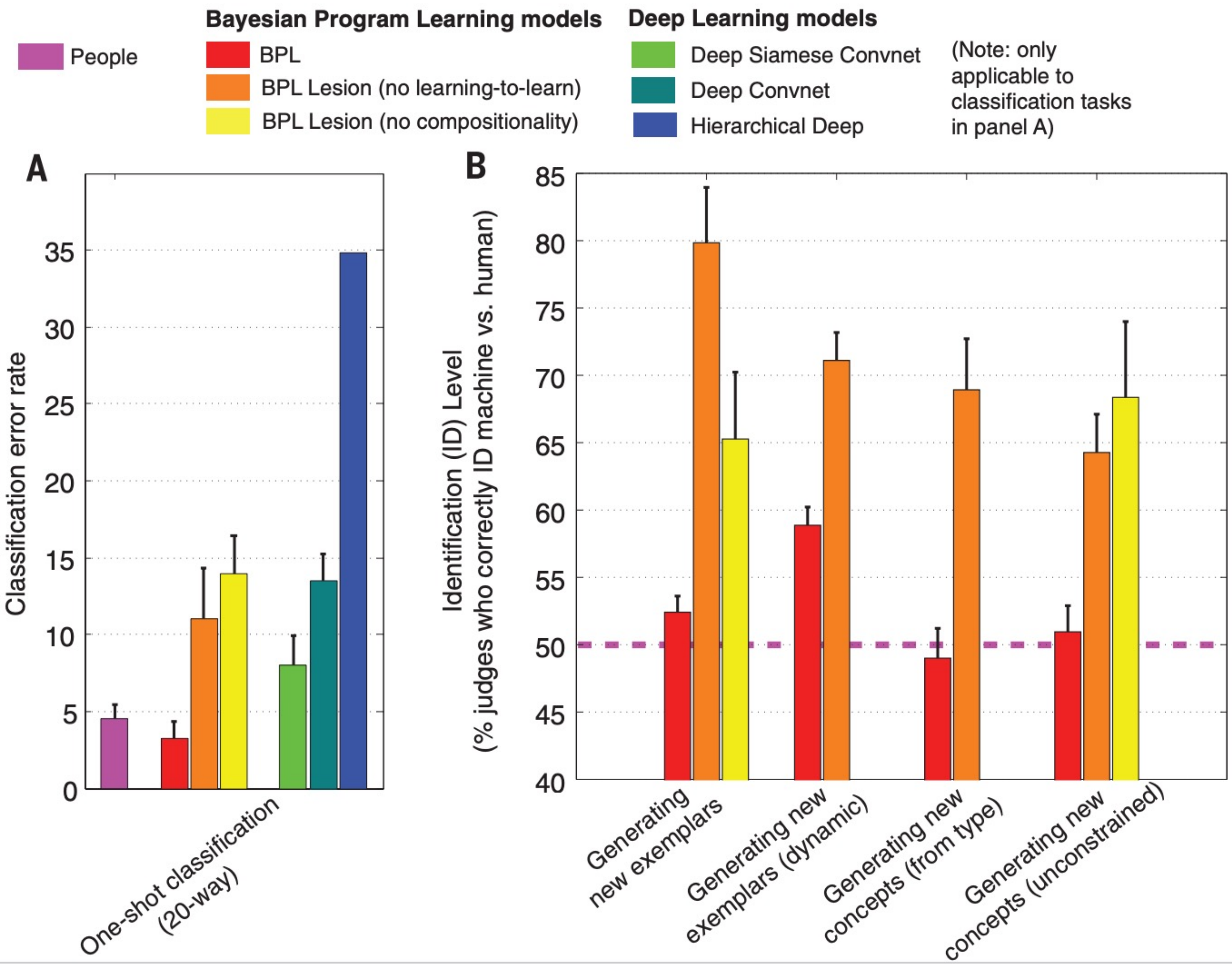
B

procedure GENERATETYPE
$\kappa \leftarrow P(\kappa)$ ▷ Sample number of parts
for $i = 1 \ldots \kappa$ **do**
$n_i \leftarrow P(n_i|\kappa)$ ▷ Sample number of sub-parts
for $j = 1 \ldots n_i$ **do**
$s_{ij} \leftarrow P(s_{ij}|s_{i(j-1)})$ ▷ Sample sub-part sequence
end for
$R_i \leftarrow P(R_i|S_1, ..., S_{i-1})$ ▷ Sample relation
end for
$\psi \leftarrow \{\kappa, R, S\}$
return @GENERATETOKEN(ψ) ▷ Return program

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 $I^{(m)}$

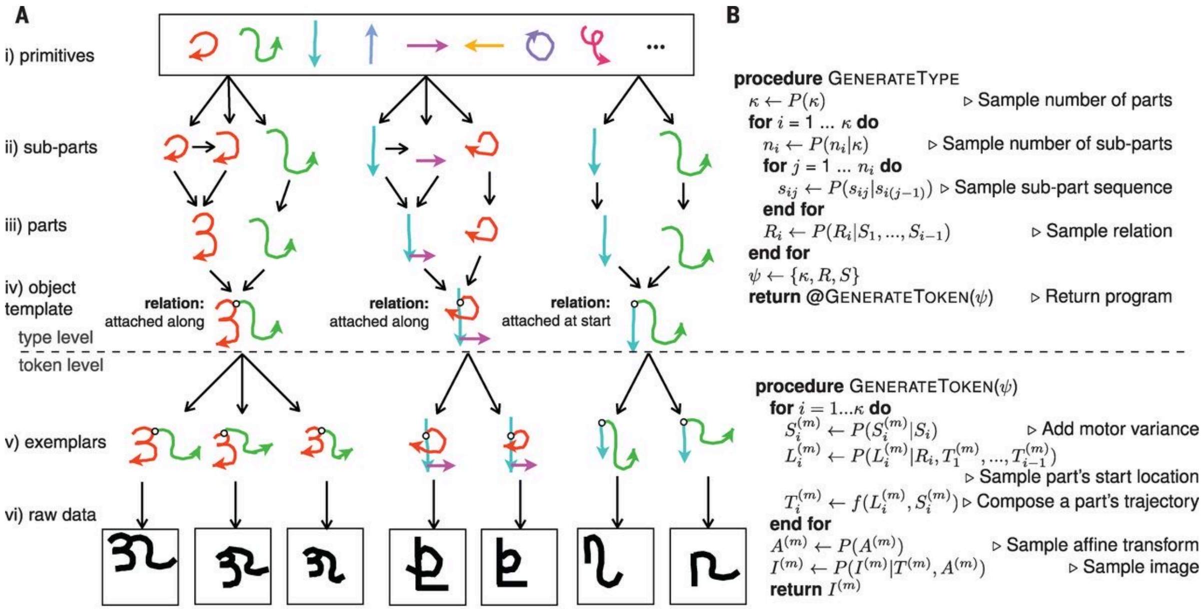
Inference. Given a character, infer generation

Human Level. And More!

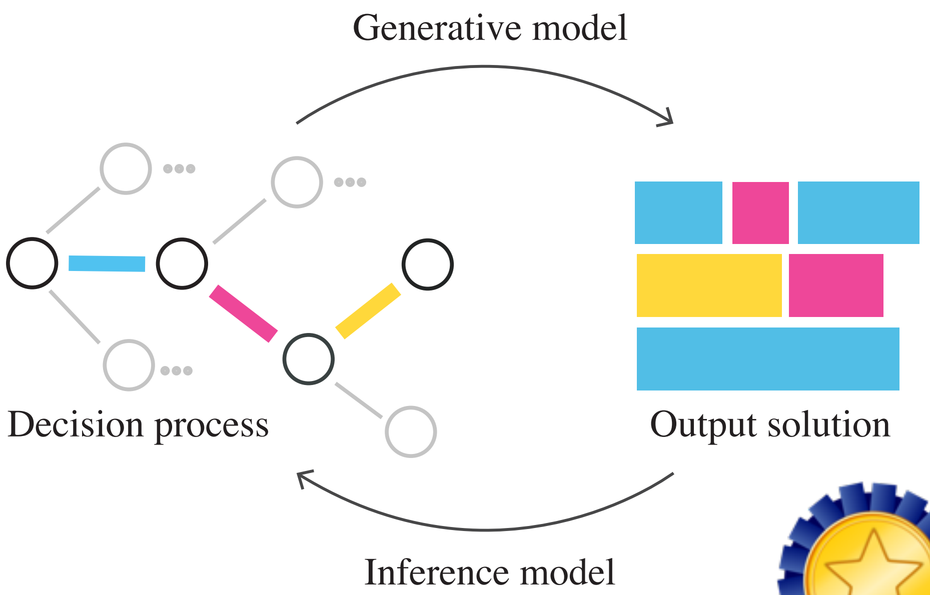


Generative Model of Grading

Lake et al, 2015

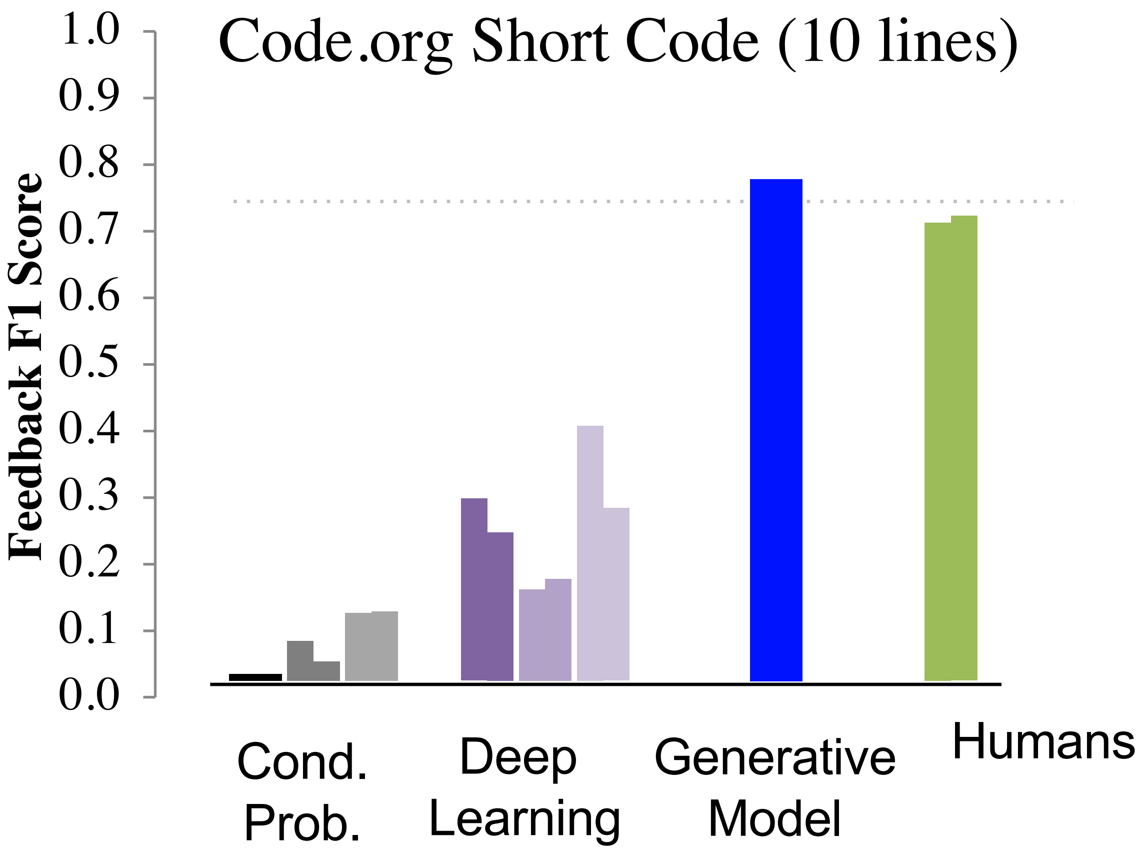


Muke Wu, Ali Malik, Noah Goodman, Chris Piech, 2019

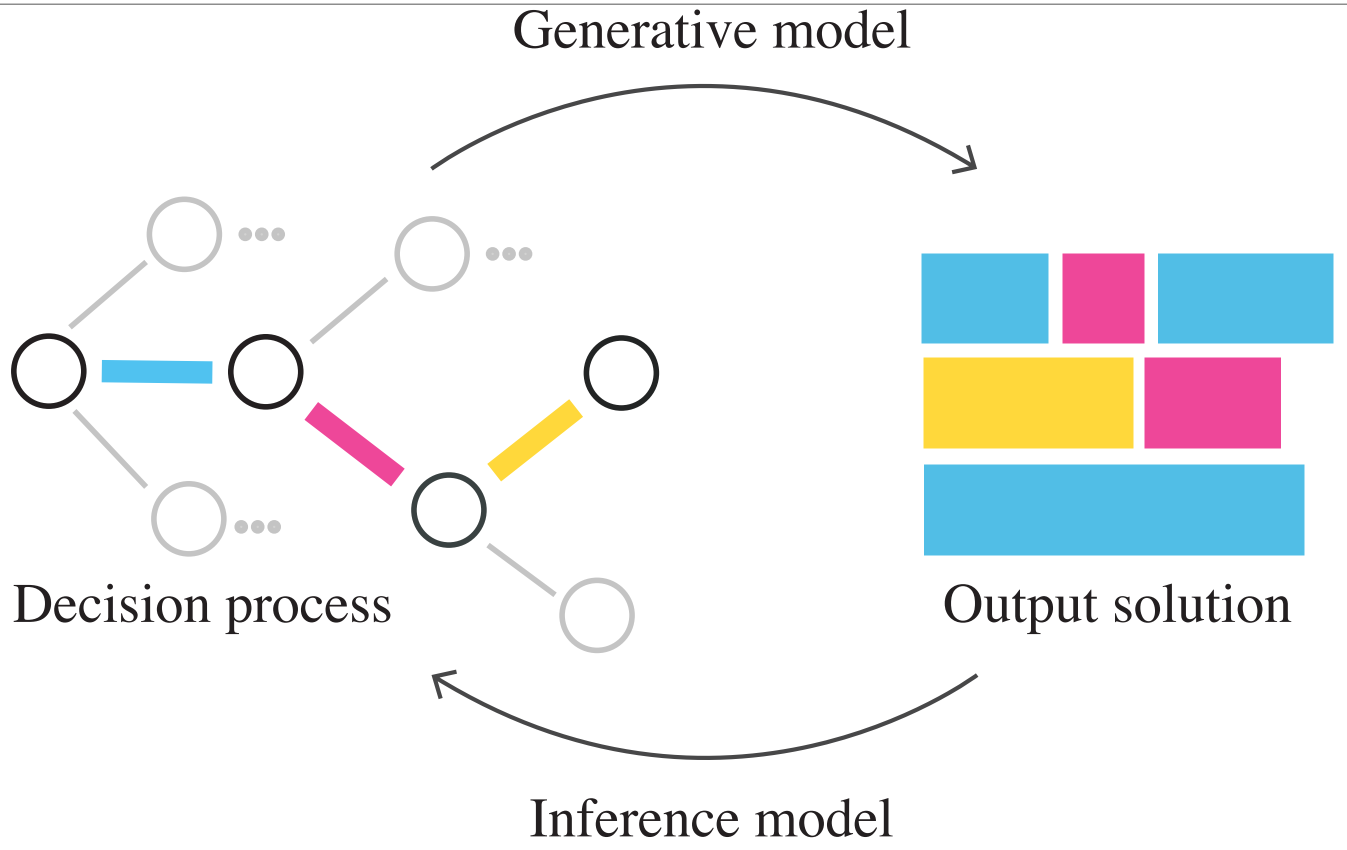


Outstanding Student paper award, AAAI 2019

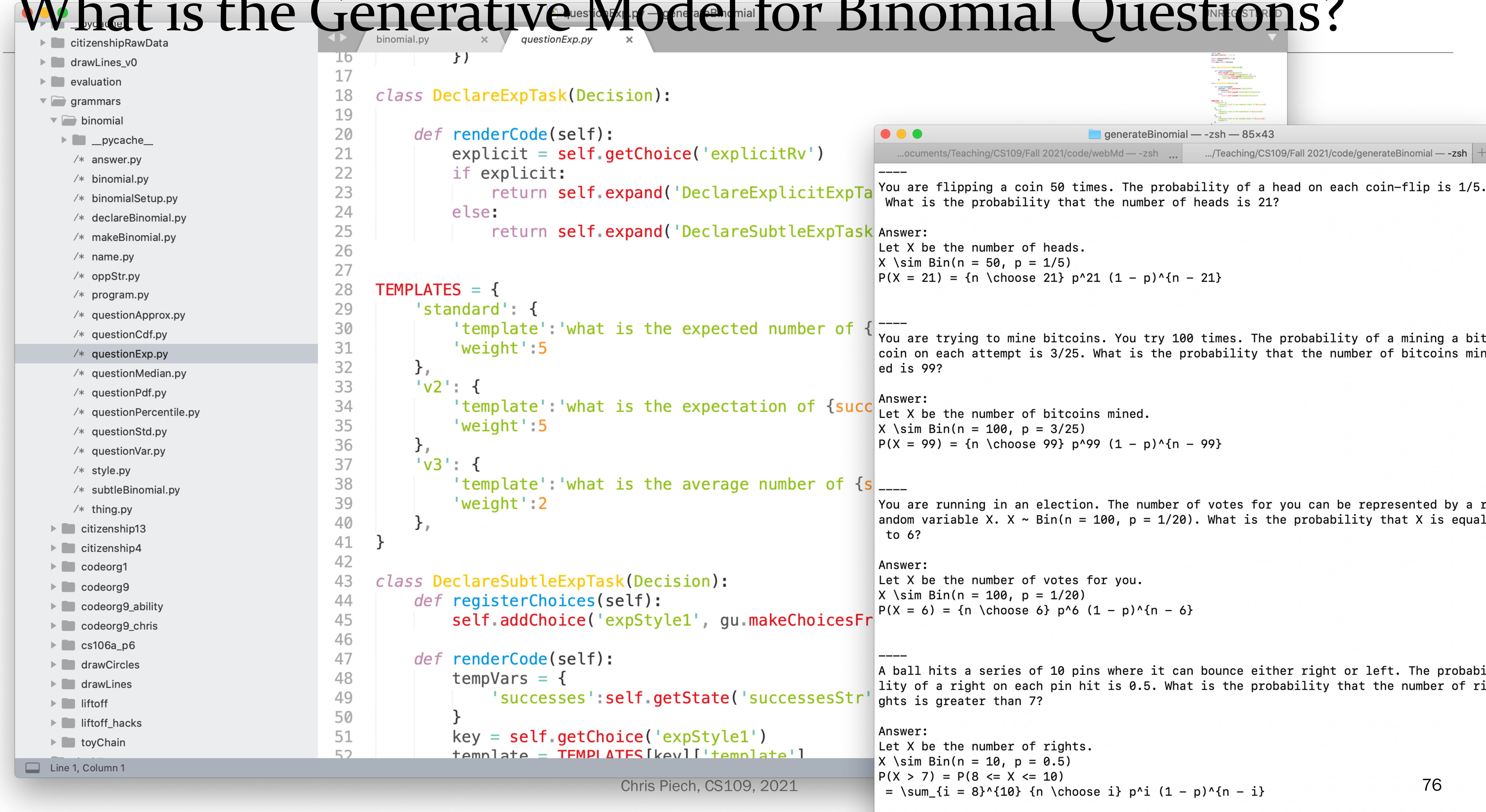
Computers Couldn't Understand Code



Idea 2 Text


Generative model
Decision process
Output solution
Inference model

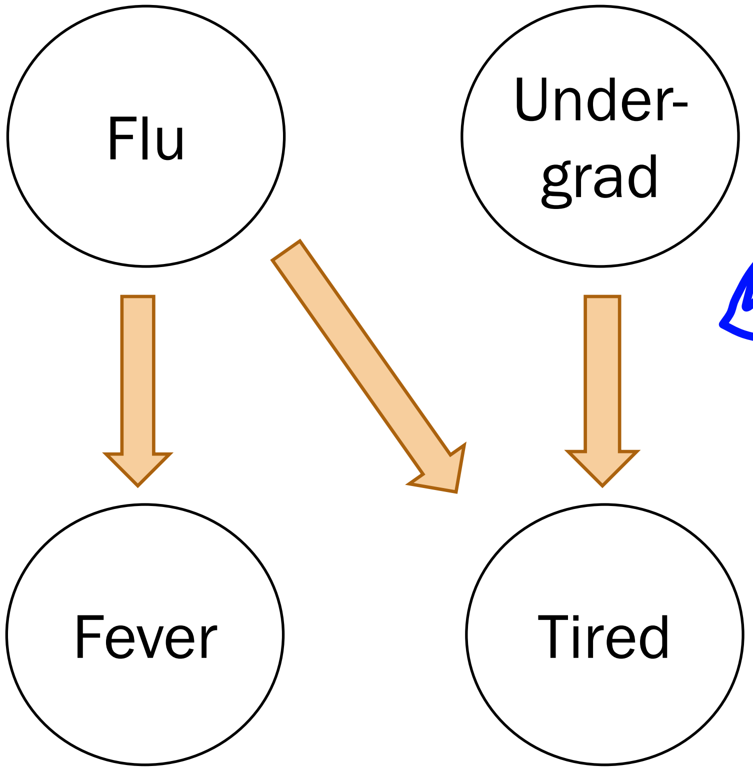
What is the Generative Model for Binomial Questions?



What haven’t we talked about?

Inference via math

$P(F_{lu} = 1) = 0.1$ $P(U = 1) = 0.8$



1. Learn this from data

2. Learn this from data

$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

See you Friday

One of my favorite classes…