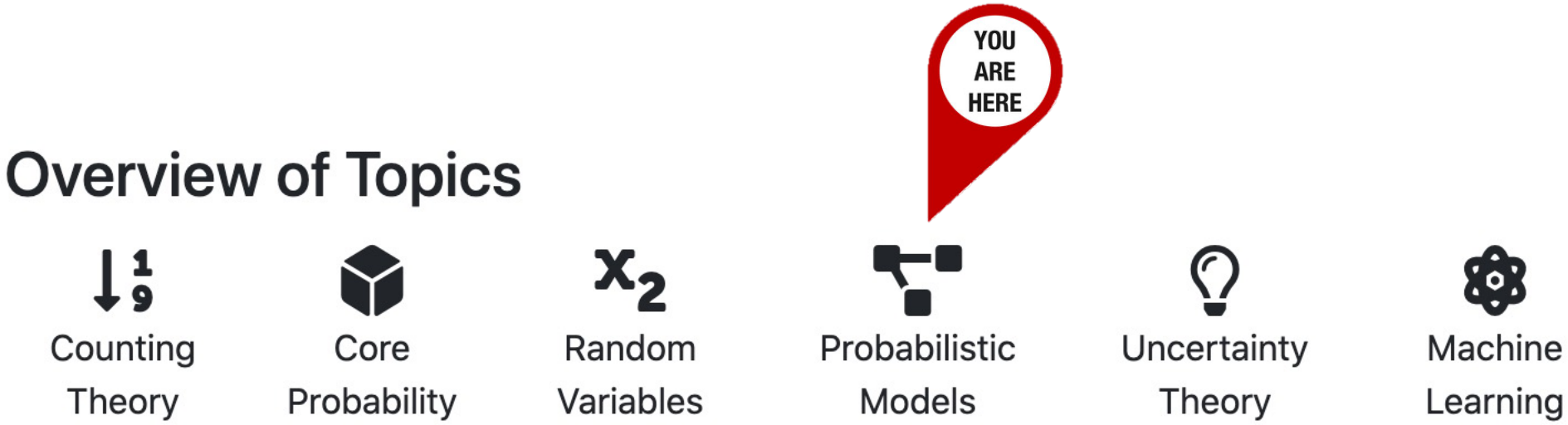
General Inference

Chris Piech
CS109, Stanford University

Announcements

- Midterm next Tuesday.
  - YOU👏 ARE👏 GOING👏 TO👏 BE👏 AWESOME👏
- Course Reader part3
- No class on Monday
- Lecture re-recording from Wednesday
- Friday

Where are we in CS109?



Where are we locally?

Discrete Models:
General Case,
Multinomial

Continuous Models:
General Case,
Multi-Gauss

Inference
Conclusions
from
Observations

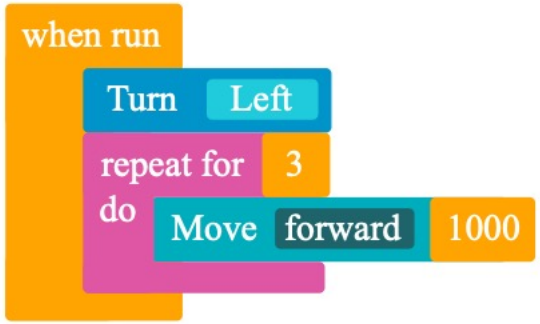
Modelling:
Make your own!

General Inference:
Use computers
to infer

Computers Couldn't Understand Code

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

Challenge

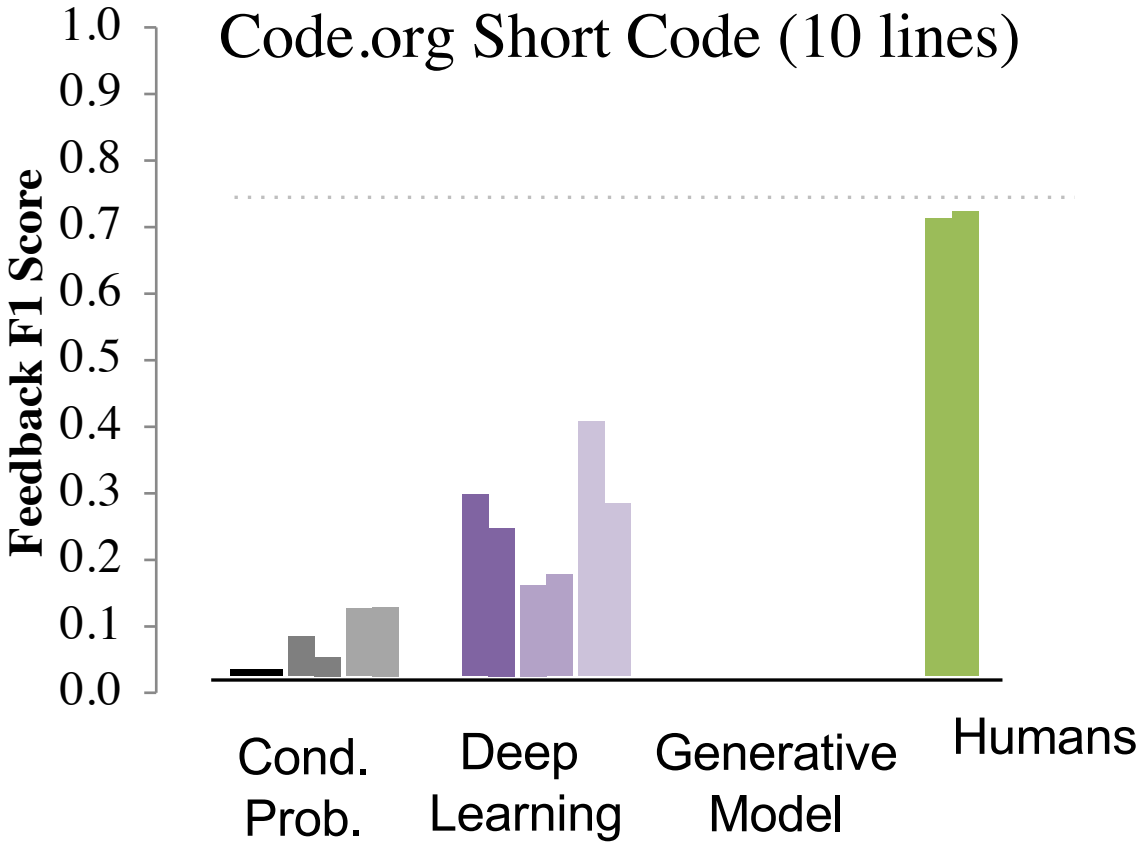


Student Code

You need to move and turn in your loop

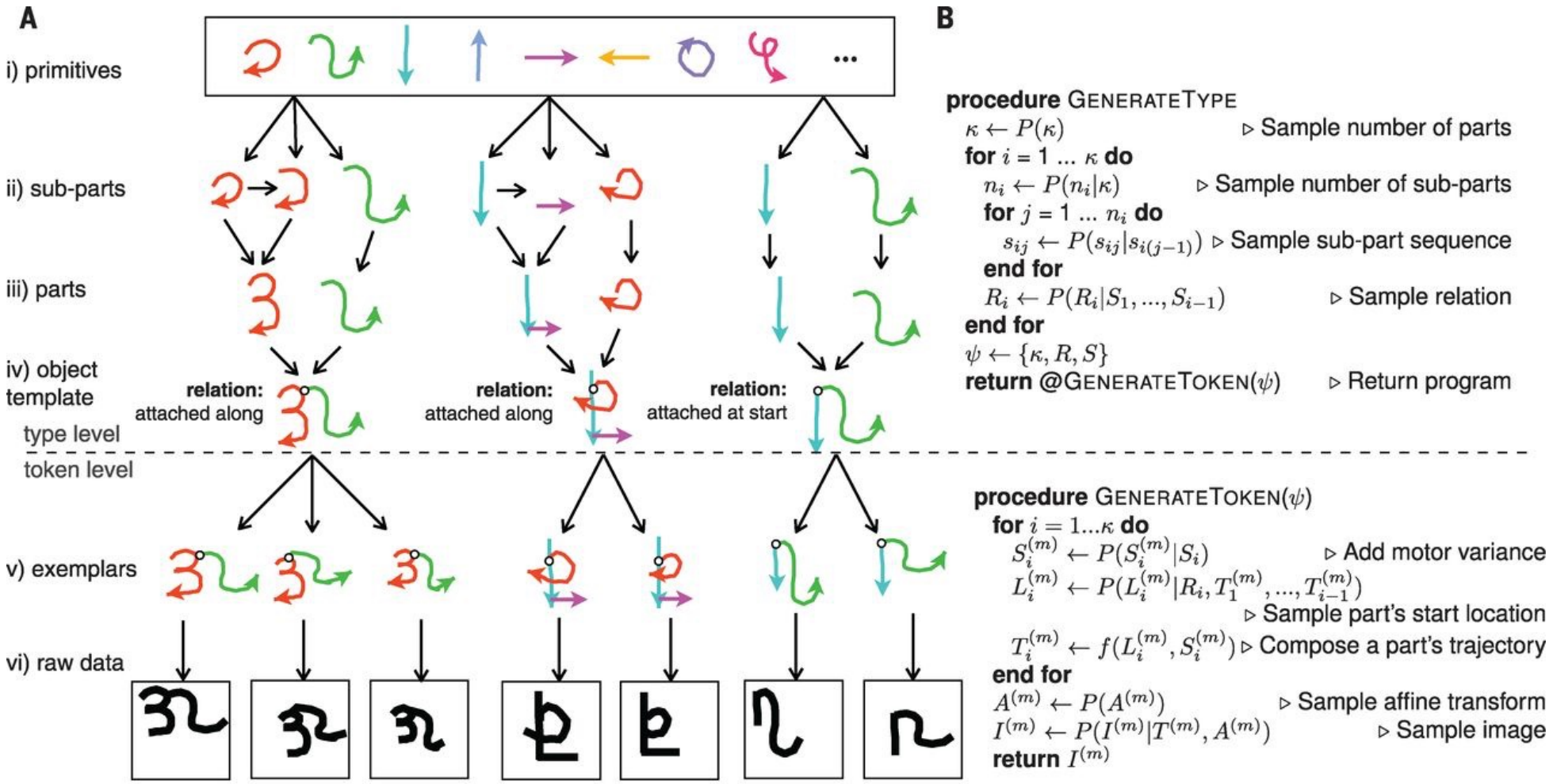
Insight

Computers Couldn't Understand Code

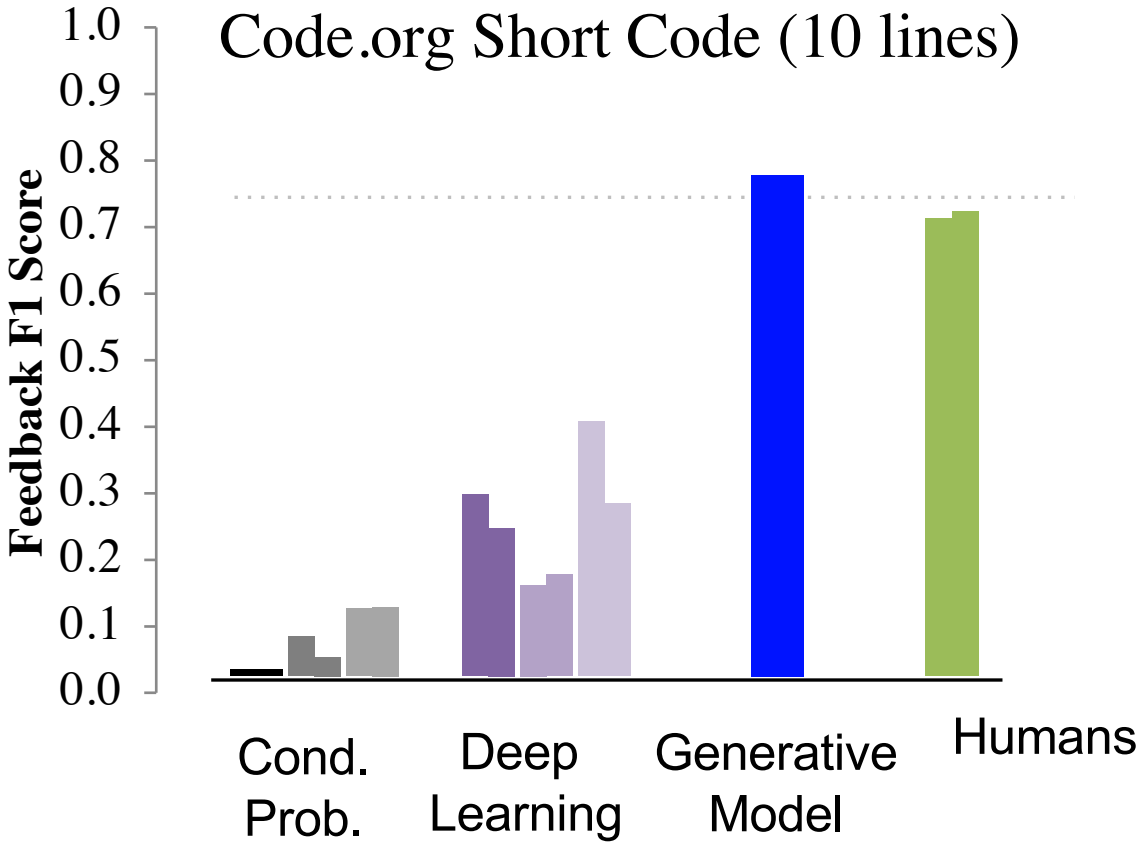


Generative Model of Characters

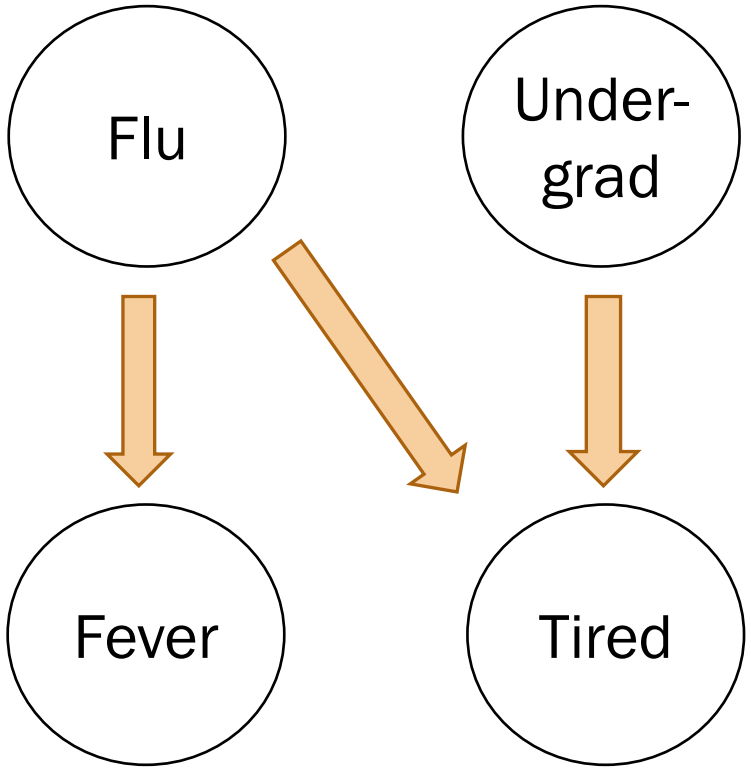
Lake et al, 2015



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,

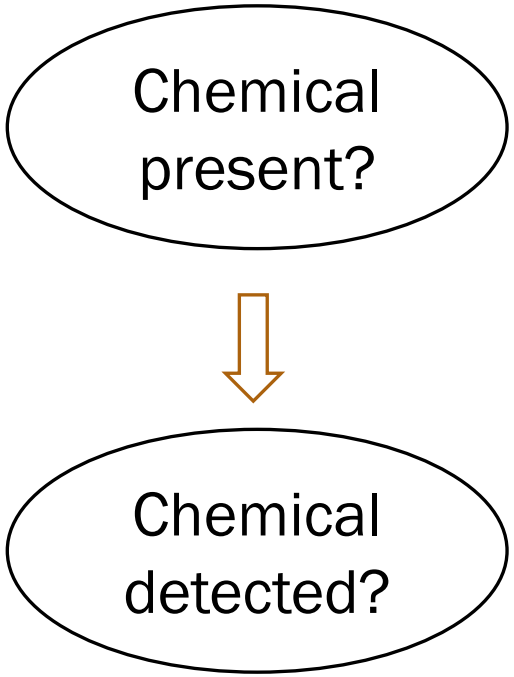
Each random variable is caused by its parents. Def P(node | parents)

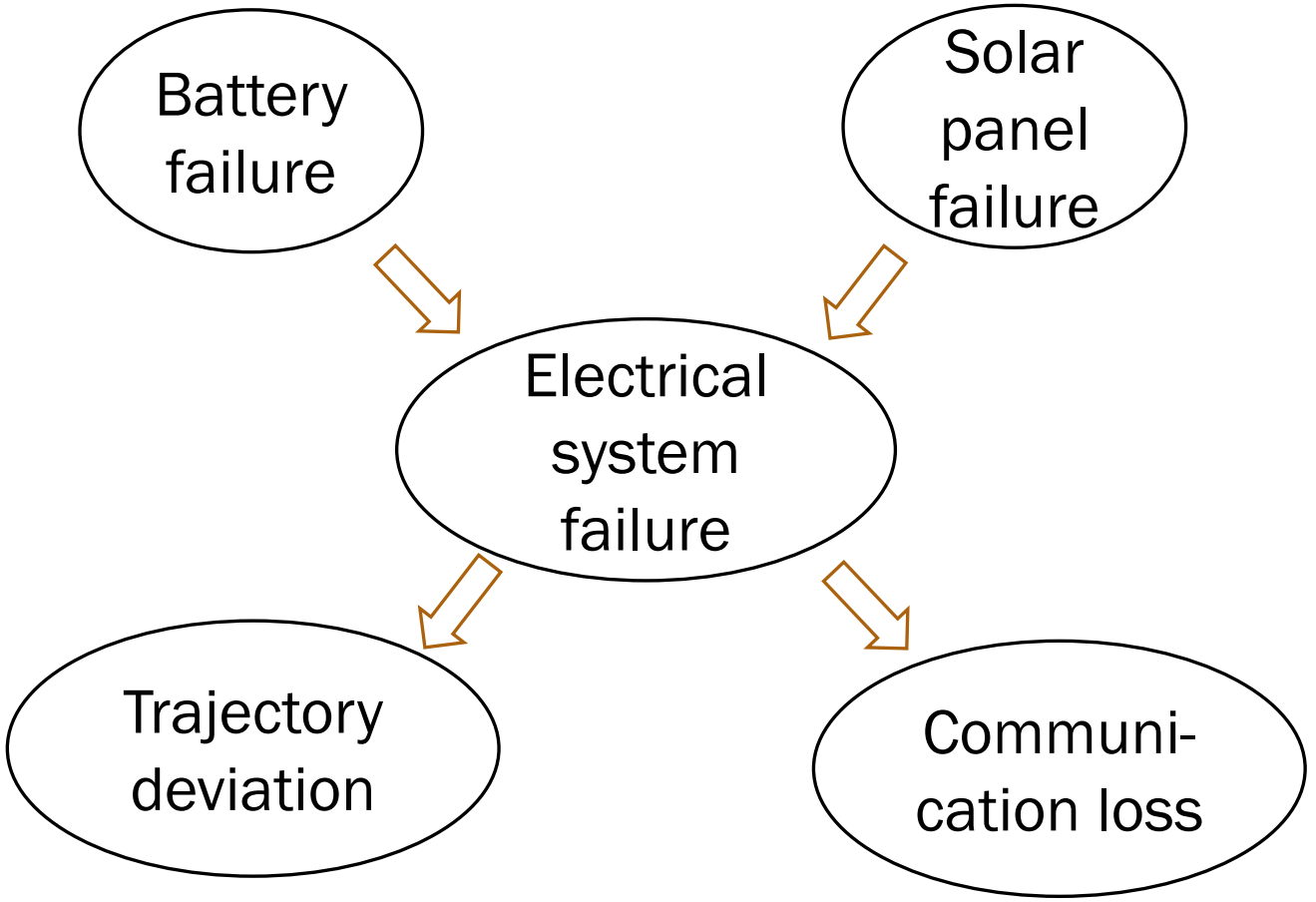
- Node: random variable
- Directed edge: causality

Examples:

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

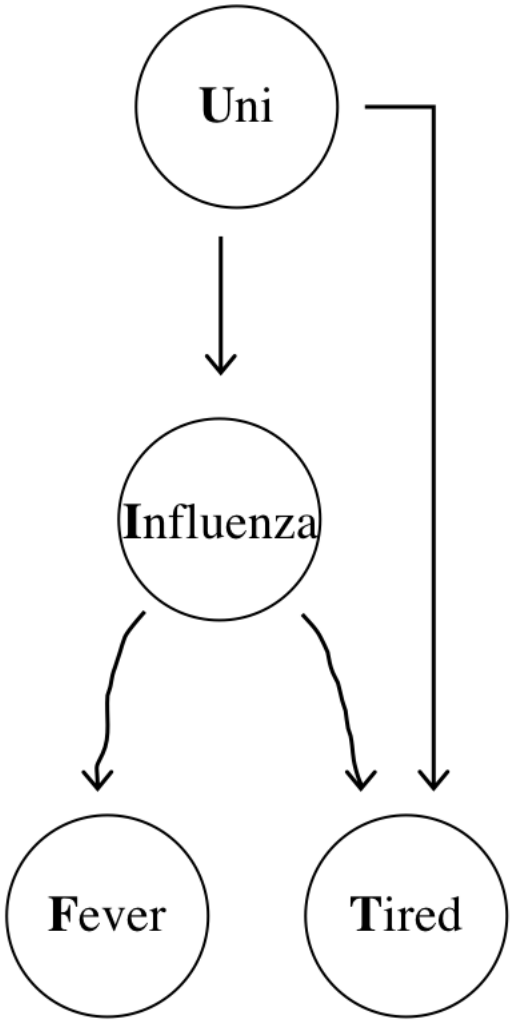
Other applications





Bayesian Network

Simple Disease Model

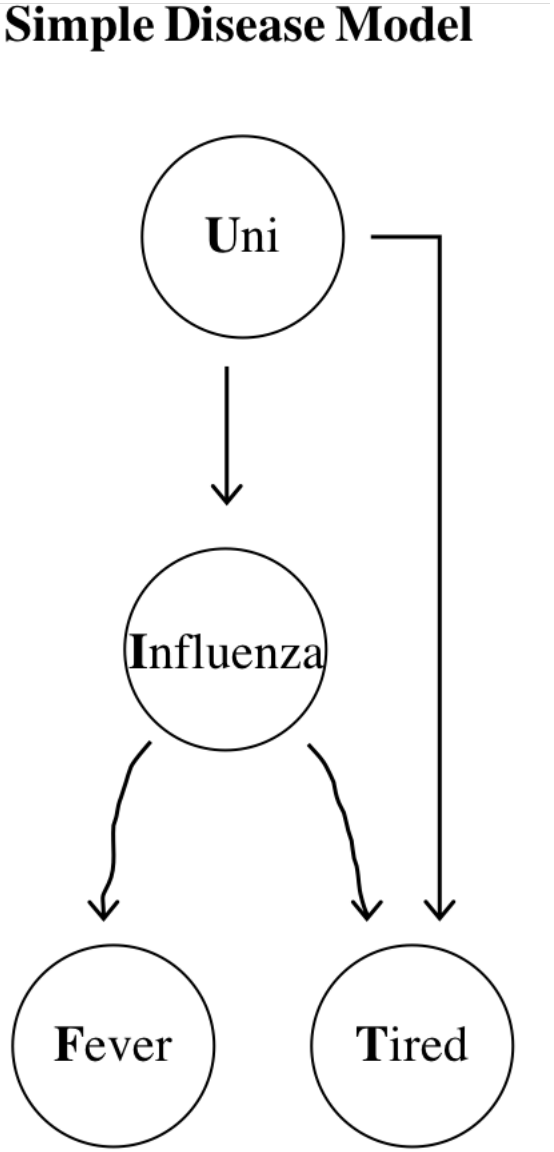



```
def get_prob_Xi(x, parents):
    # what is the probability that Xi = x
    # given the list parents of assignments to
    # the parents variables Xi
```

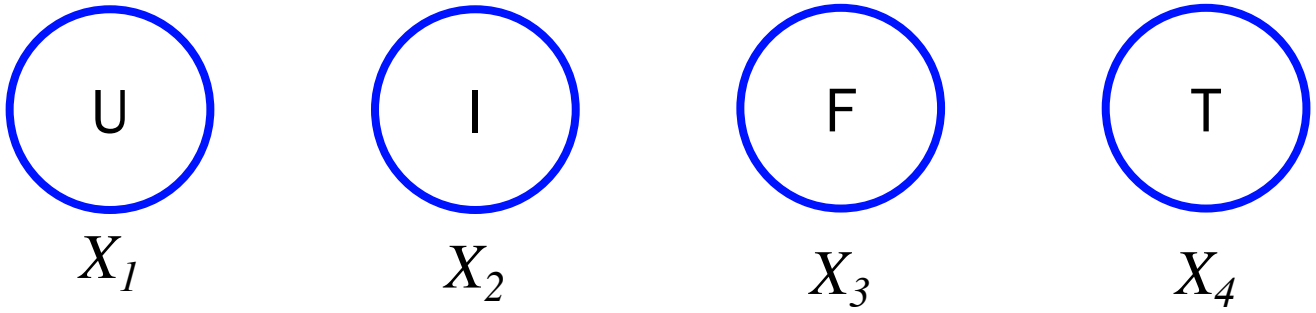
$$P(\text{Uni} = 1) = 0.8$$
$$P(\text{Influenza} = 1 | \text{Uni} = 1) = 0.2$$
$$P(\text{Influenza} = 1 | \text{Uni} = 0) = 0.1$$
$$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 0) = 0.1$$
$$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 0) = 0.8$$

$$P(\text{Fever} = 1 | \text{Influenza} = 1) = 0.9$$
$$P(\text{Fever} = 1 | \text{Influenza} = 0) = 0.05$$
$$P(\text{Tired} = 1 | \text{Uni} = 0, \text{Influenza} = 1) = 0.9$$
$$P(\text{Tired} = 1 | \text{Uni} = 1, \text{Influenza} = 1) = 1.0$$

Bayesian Network Assumption



Order nodes by ancestry

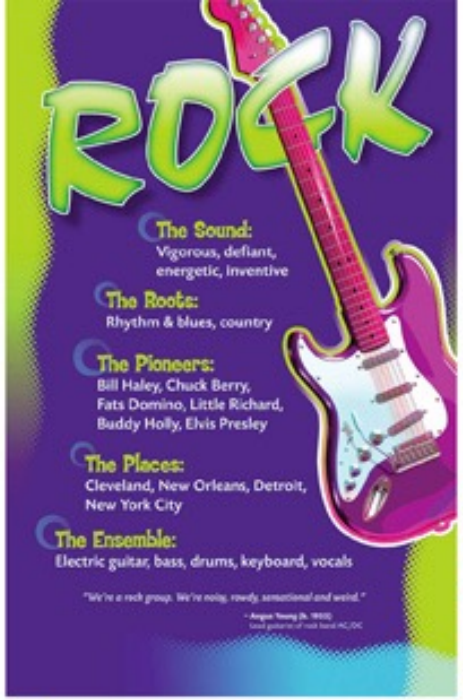


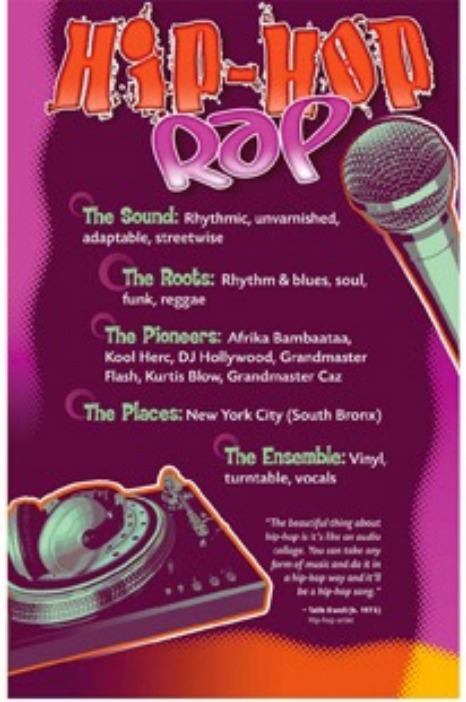
$$P(\text{Joint}) = \prod_i P(x_i | x_{i-1}, \ldots, x_1)$$

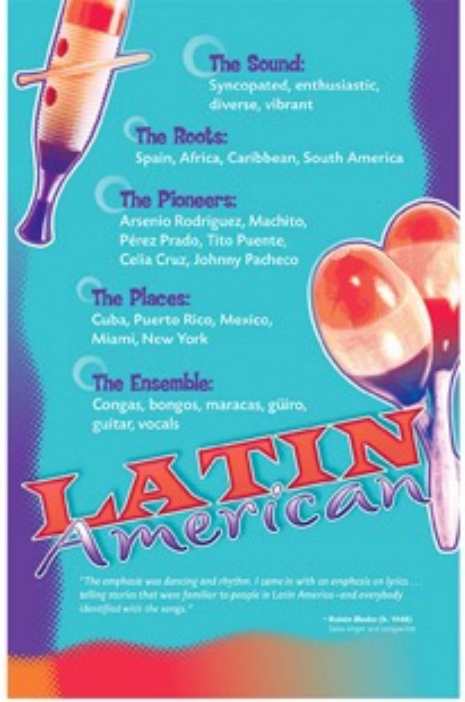
$$= \prod_i P(x_i | \text{Values of parents of } X_i)$$

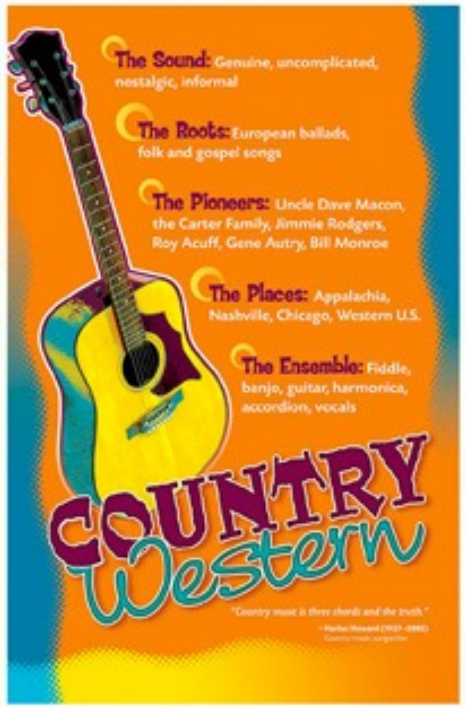
Assume: Once you know the value of the parents of a variable in your network, X_i, any further information about non-descendents will not change your belief in X_i.

How do people design these?









AutoSave OFF — Search Sheet

Home | Insert | Page Layout | Formulas | Data | Share

Clipboard | Font | Alignment | Number | Conditional Formatting | Format as Table | Cell Styles | Cells | Editing

C15 | fx | 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

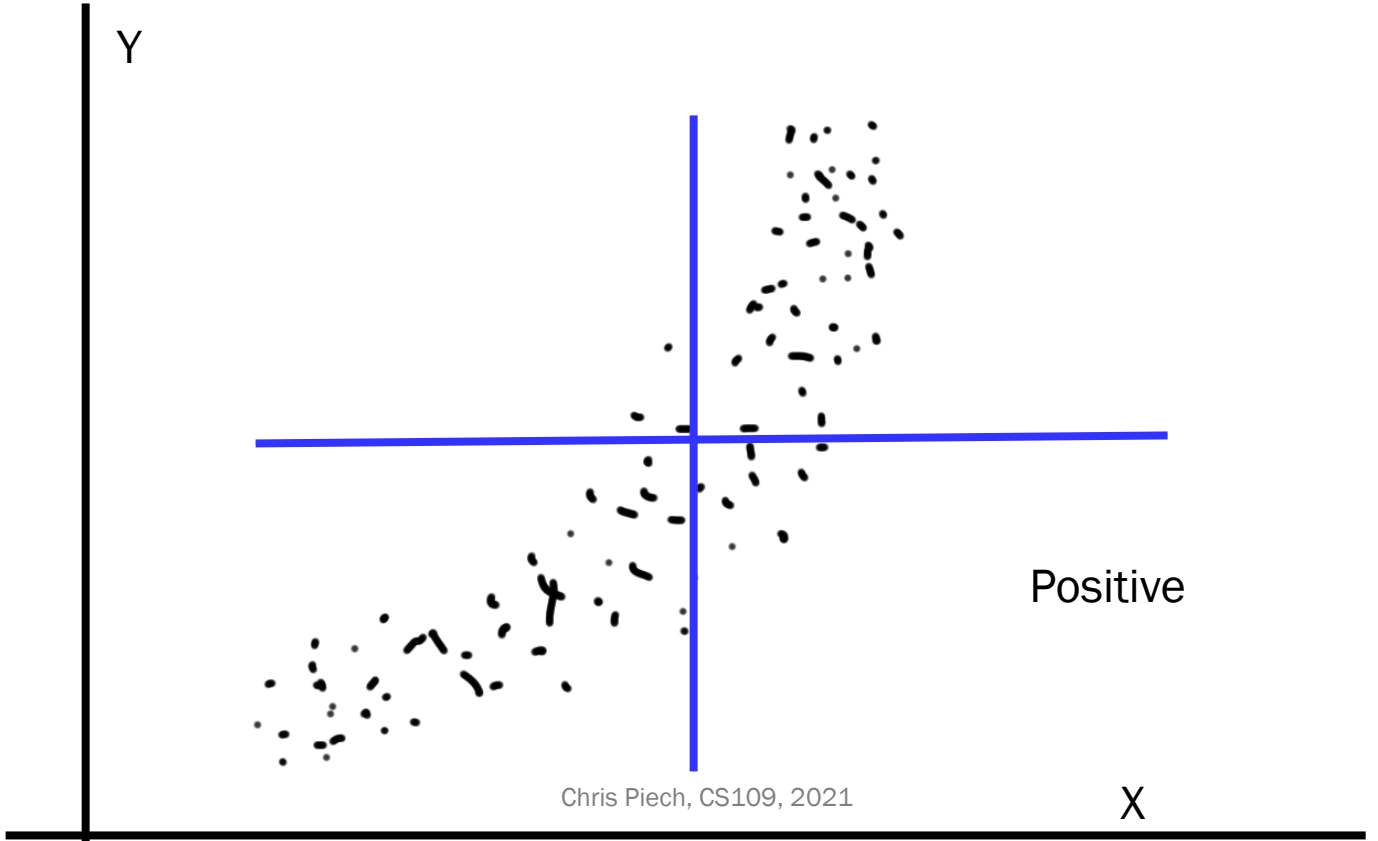
music

Ready — 100%

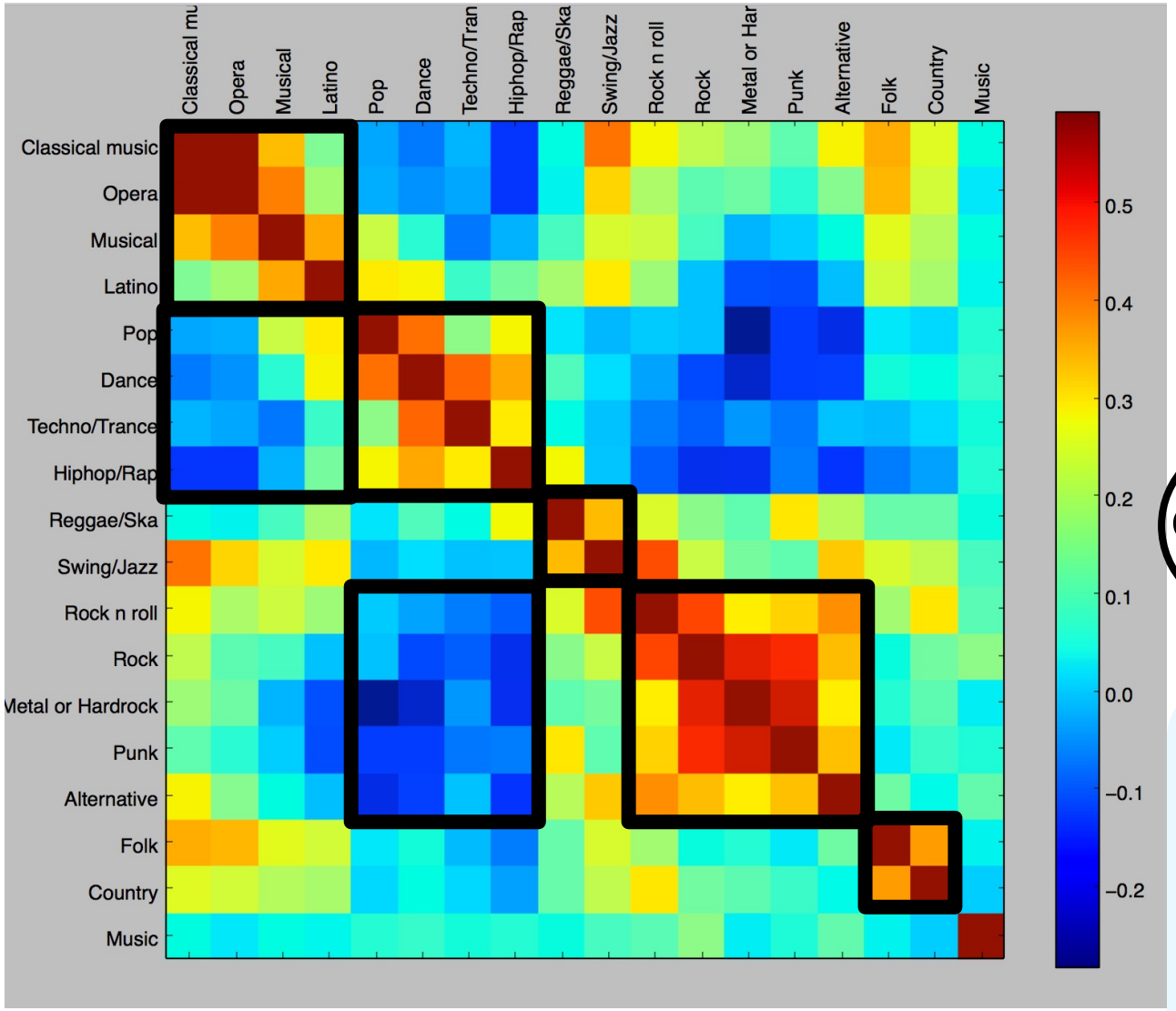
Calculate the Covariance (new stat!)

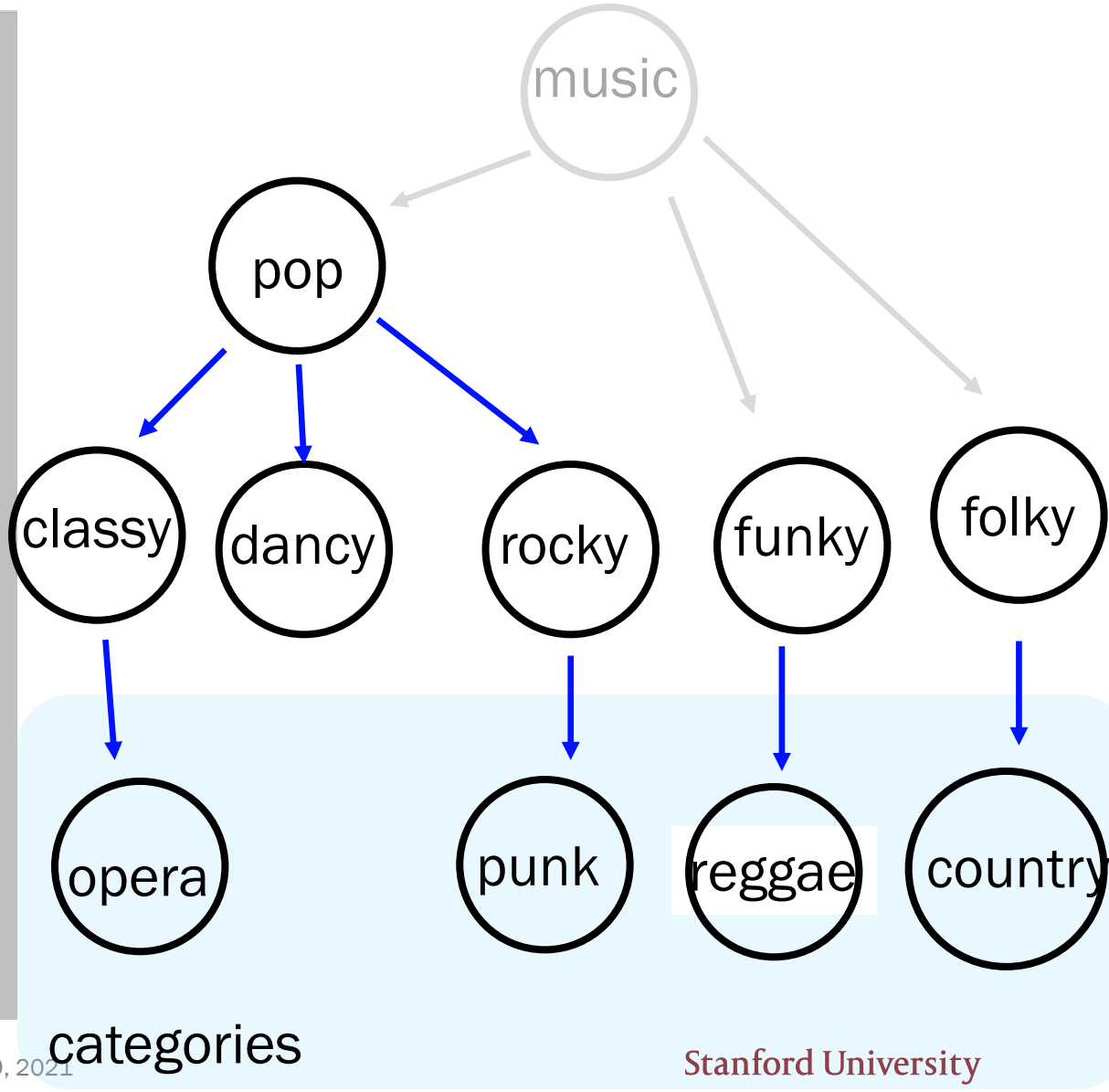
$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

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

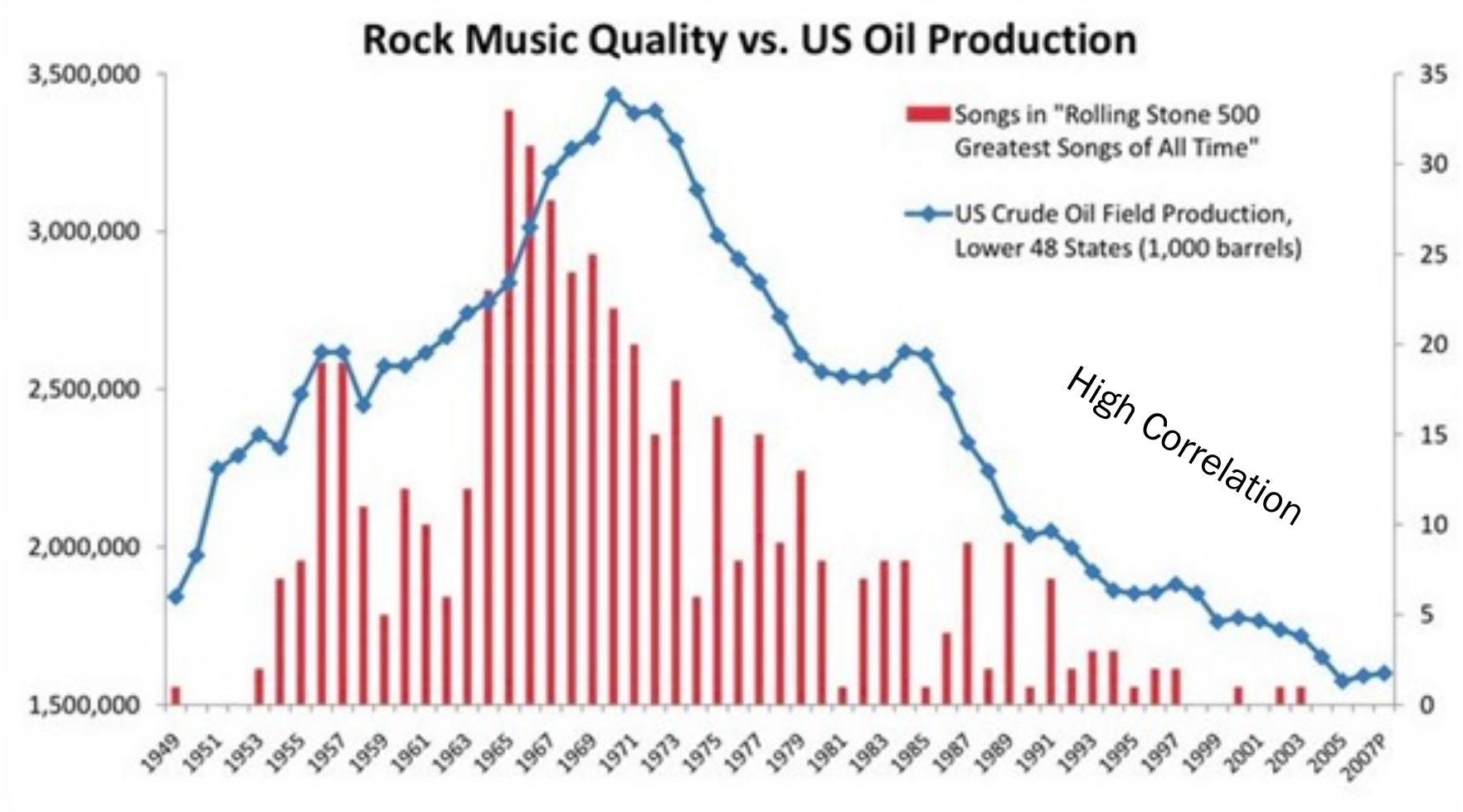


From Correlation to Bayes Net. Alternative!





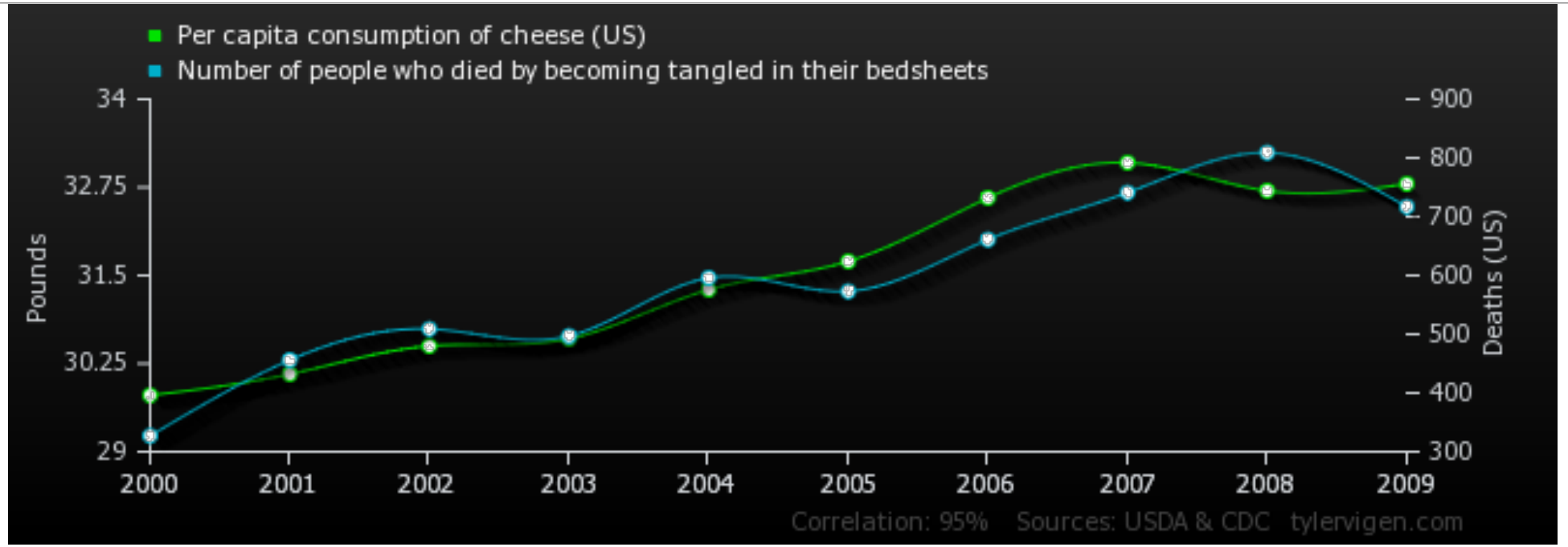
Rock Music Vs Oil?



Hubbert Peak Theory

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

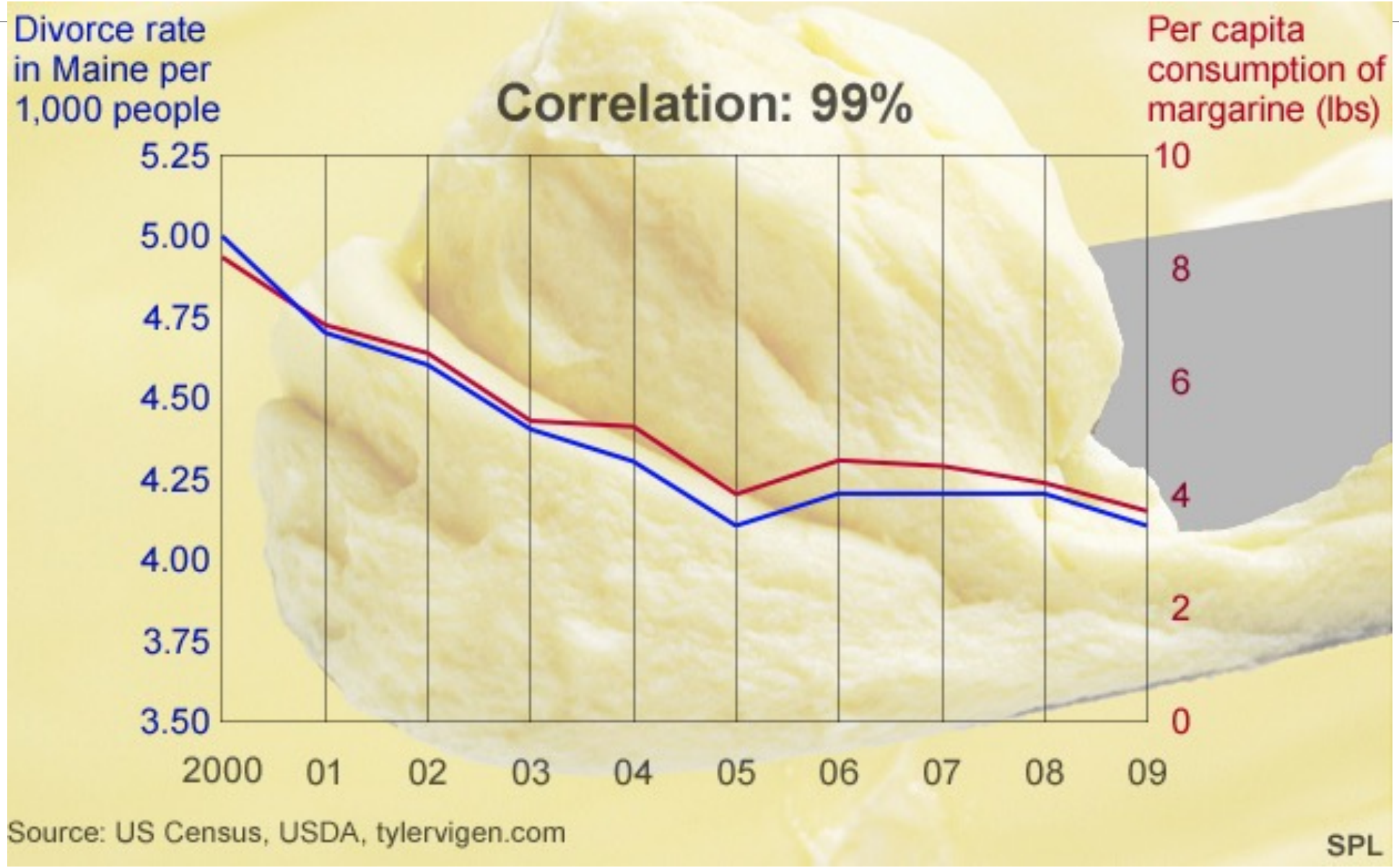
Tell your friends!



	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

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

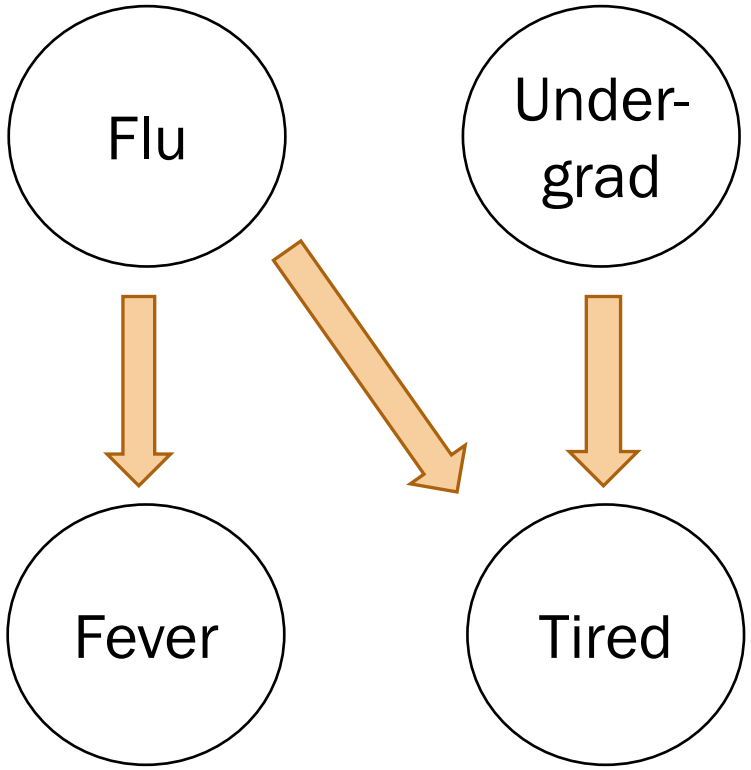
$$p_{N|X}(n|x) = \frac{f_{X|N}(x|n)p_N(n)}{f_X(x)}$$

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

So it begins…

Inference: Algebra

Bayes Nets: Conditional independence



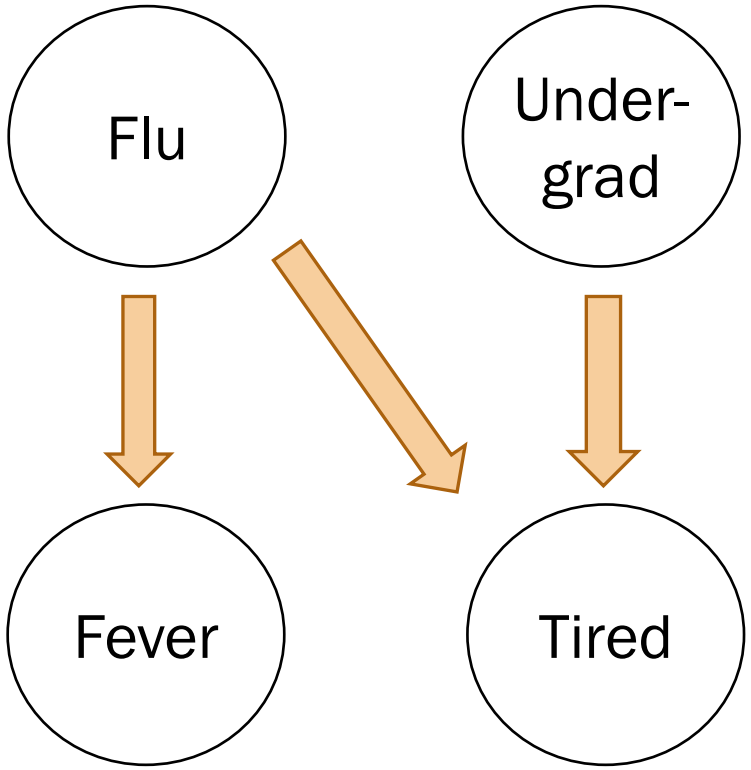
In a Bayesian Network,

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

- Node: random variable
- Directed edge: conditional dependency

Inference via math

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



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

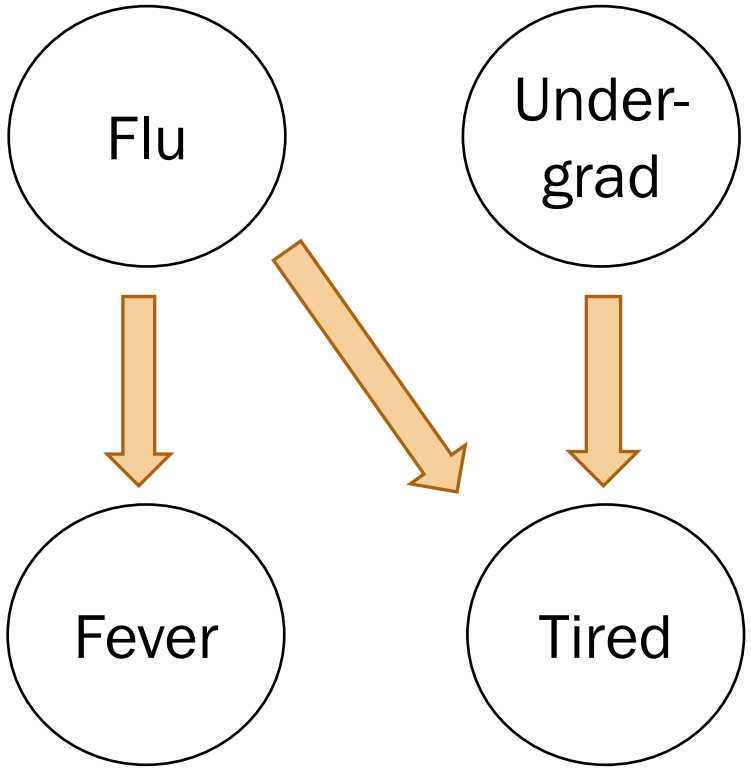
Compute joint probabilities using chain rule.

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

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

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

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

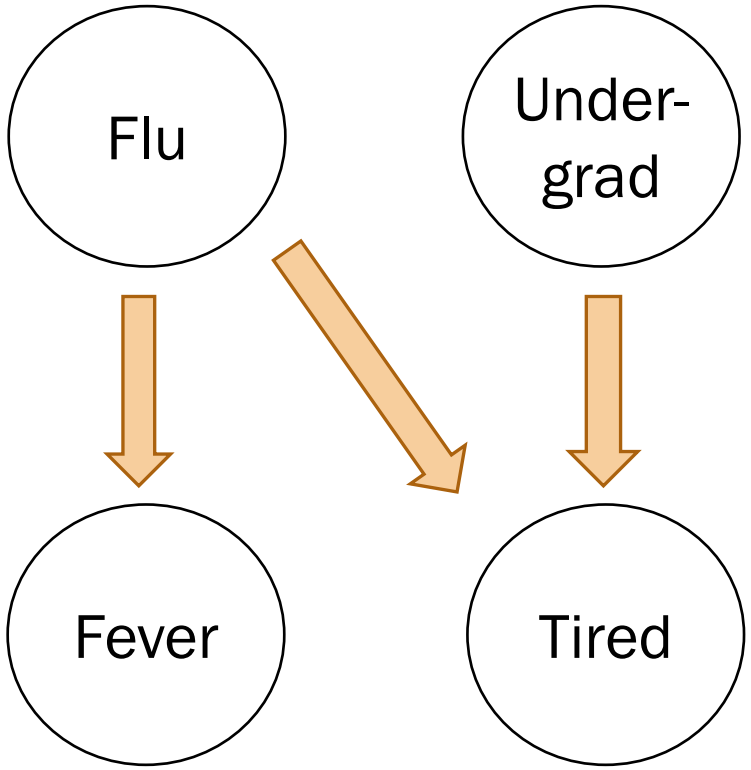
2. Definition of conditional probability

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

$= 0.095$

Inference via math

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



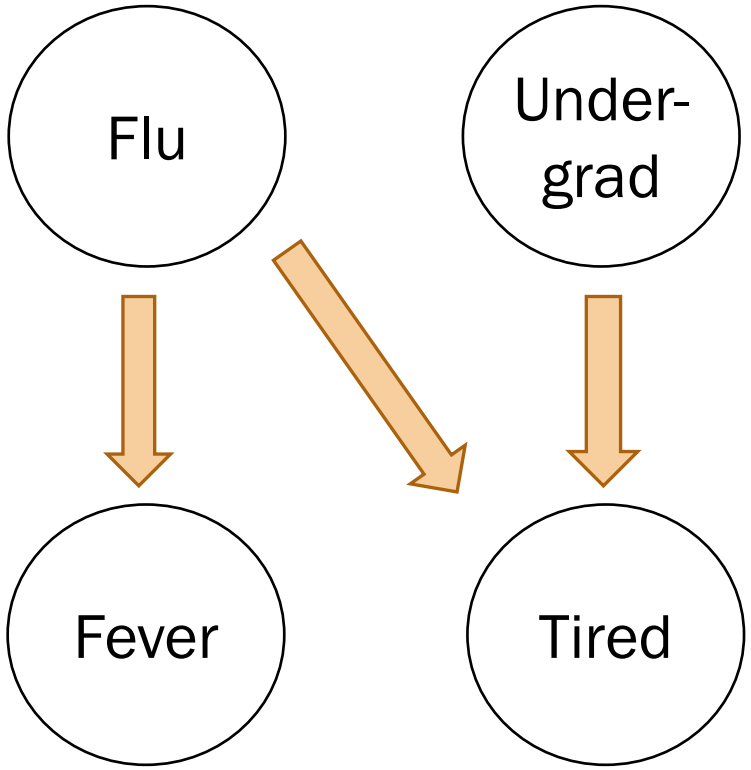
3. $P(F_{lu} = 1 | U = 1, T = 1)$?

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

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

Inference via math

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



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

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

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

1. Compute joint probabilities

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

2. Definition of conditional probability

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

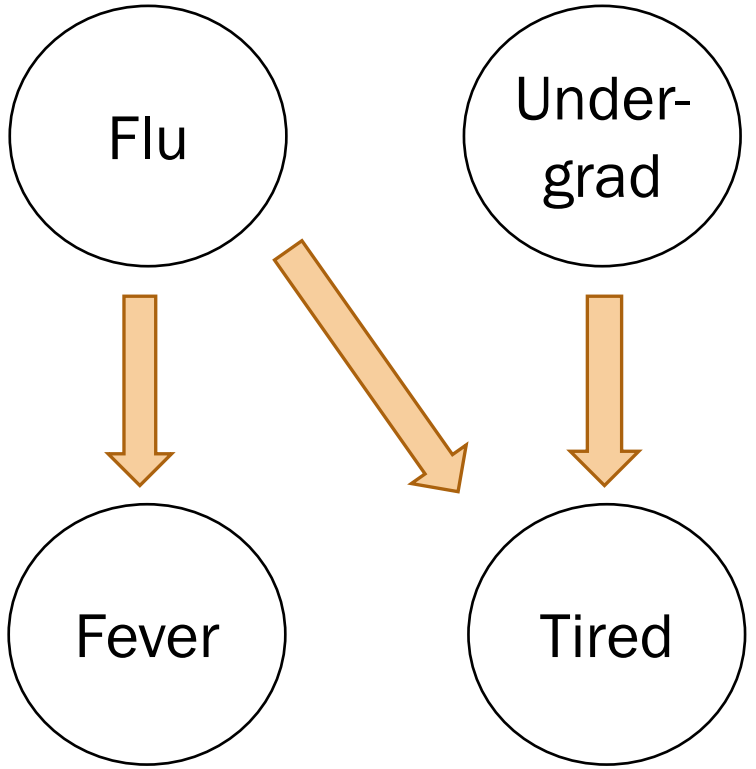
$= 0.122$

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

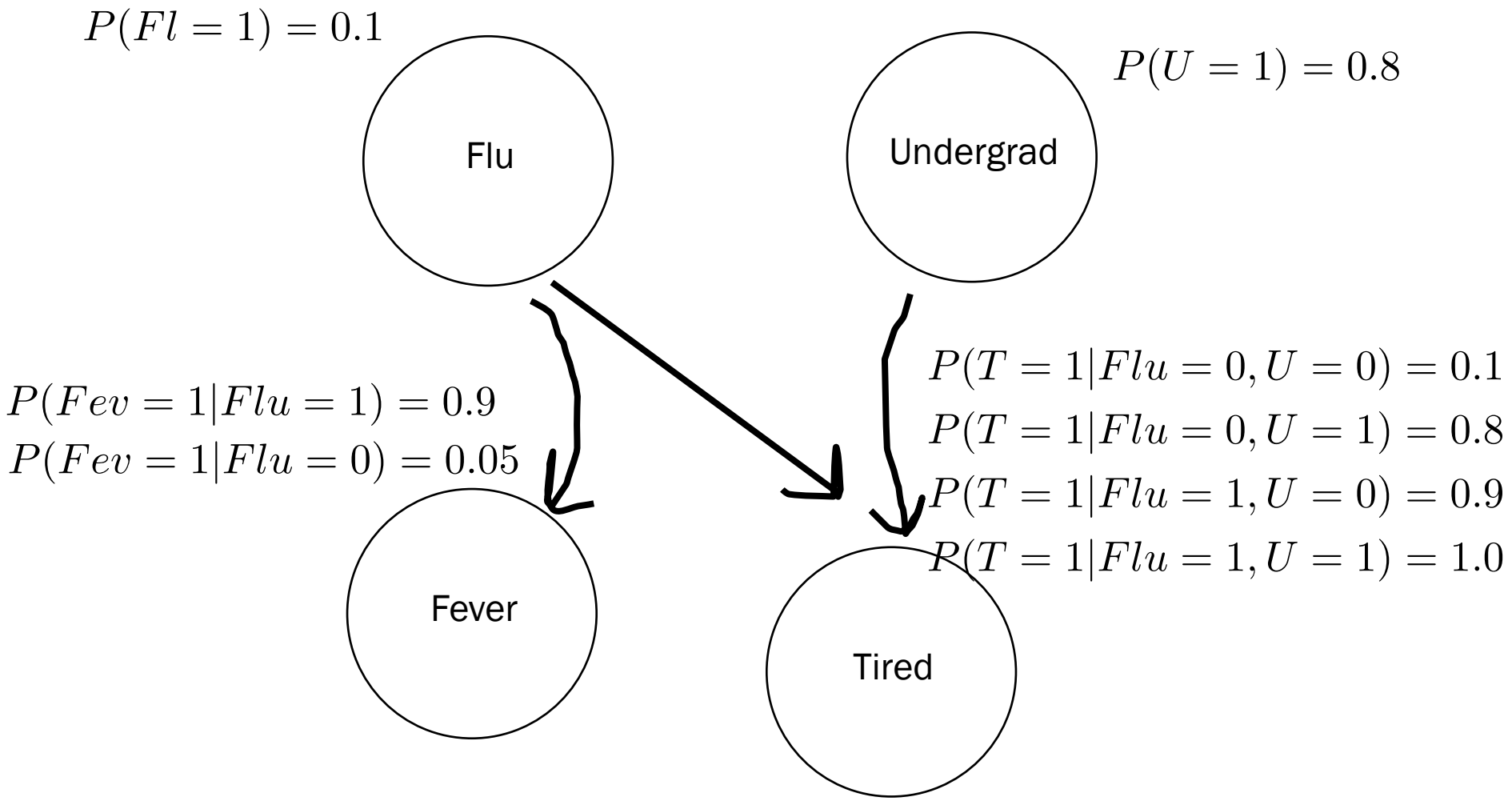
Too many possible inference questions one could ask...

Alg #1: Rejection Sampling

```
3  N_SAMPLES = 100000
4
5  # Program: Joint Sample
6  # ----------------------
7  # we can answer any probability question
8  # with multivariate samples from the joint,
9  # where conditioned variables match
10 def main():
11     obs = getObservation()
12     print 'Observation = ', obs
13
14     samples = sampleATon()
15     prob = probFluGivenObs(samples, obs)
16     print 'Pr(Flu) = ', prob
```

```
71  # Method: Sample A Ton
72  # ---------------------
73  # chose N_SAMPLES with likelihood proportional
74  # to the joint distribution
75  def sampleATon():
76      samples = []
77      for i in range(N_SAMPLES):
78          sample = makeSample()
79          samples.append(sample)
80      return samples
```

Recall: Probabilistic Model



```
82  # Method: Make Sample
83  # ------------------
84  # chose a single sample from the joint distribut
85  # based on the medical "Probabilistic Graphical M
86  def makeSample():
87      # prior on causal factors
88      flu = bern(0.1)
89      und = bern(0.8)
90
91      # choose fever based on flue
92      if flu == 1: fev = bern(0.9)
93      else:        fev = bern(0.05)
94
95      # choose tired based on (undergrade and flu)
96      if und == 1 and flu == 1:   tir = bern(1.0)
97      elif und == 1 and flu == 0: tir = bern(0.8)
98      elif und == 0 and flu == 1: tir = bern(0.9)
99      else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82  # Method: Make Sample
83  # ------------------
84  # chose a single sample from the joint distribut
85  # based on the medical "Probabilistic Graphical M
86  def makeSample():
87      # prior on causal factors
88      flu = bern(0.1)
89      und = bern(0.8)
90
91      # choose fever based on flue
92      if flu == 1: fev = bern(0.9)
93      else:        fev = bern(0.05)
94
95      # choose tired based on (undergrade and flu)
96      if und == 1 and flu == 1:   tir = bern(1.0)
97      elif und == 1 and flu == 0: tir = bern(0.8)
98      elif und == 0 and flu == 1: tir = bern(0.9)
99      else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82  # Method: Make Sample
83  # ------------------
84  # chose a single sample from the joint distribut
85  # based on the medical "Probabilistic Graphical M
86  def makeSample():
87      # prior on causal factors
88      flu = bern(0.1)
89      und = bern(0.8)
90
91      # choose fever based on flue
92      if flu == 1: fev = bern(0.9)
93      else:        fev = bern(0.05)
94
95      # choose tired based on (undergrade and flu)
96      if und == 1 and flu == 1:   tir = bern(1.0)
97      elif und == 1 and flu == 0: tir = bern(0.8)
98      elif und == 0 and flu == 1: tir = bern(0.9)
99      else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

```
82  # Method: Make Sample
83  # ------------------
84  # chose a single sample from the joint distribut
85  # based on the medical "Probabilistic Graphical M
86  def makeSample():
87      # prior on causal factors
88      flu = bern(0.1)
89      und = bern(0.8)
90
91      # choose fever based on flue
92      if flu == 1: fev = bern(0.9)
93      else:        fev = bern(0.05)
94
95      # choose tired based on (undergrade and flu)
96      if und == 1 and flu == 1:   tir = bern(1.0)
97      elif und == 1 and flu == 0: tir = bern(0.8)
98      elif und == 0 and flu == 1: tir = bern(0.9)
99      else:                       tir = bern(0.1)
100
101     # a sample from the joint has an
102     # assignment to *all* random variables
103     return [flu, und, fev, tir]
```

Alg #1: Rejection Sampling

```
3  N_SAMPLES = 100000
4
5  # Program: Joint Sample
6  # ----------------------
7  # we can answer any pro
8  # with multivariate sam
9  # where conditioned var
10 def main():
11     obs = getObservatio
12     print 'Observation
13
14     samples = sampleATo
15     prob = probFluGiven
16     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

```
3  N_SAMPLES = 100000
4
5  # Program: Joint Sample
6  # ----------------------
7  # we can answer any probability question
8  # with multivariate samples from the joint,
9  # where conditioned variables match
10 def main():
11     obs = getObservation()
12     print 'Observation = ', obs
13
14     samples = sampleATon()
15     prob = probFluGivenObs(samples, obs)
16     print 'Pr(Flu) = ', prob
```

```
# 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)
```

```
25  # Method: Probability of Flu Given Observation
26  # -----------------------------
27  # Calculate the probability of flu given many
28  # samples from the joint distribution and a set
29  # of ovservations to condition on.
30  def probFluGivenObs(samples, obs):
31      # reject all samples which don't align
32      # with condition
33      keepSamples = []
34      for sample in samples:
35          if checkObsMatch(sample, obs):
36              keepSamples.append(sample)
37
38      # from remaining, simply count...
39      fluCount = 0
40      for sample in keepSamples:
41          [flu, und, fev, tir] = sample
42          if flu == 1:
43              fluCount += 1
44
45      # counting can be so sweet...
46      return float(fluCount) / len(keepSamples)
```

```
25  # Method: Probability of Flu Given Observation
26  # -----------------------------
27  # Calculate the probability of flu given many
28  # samples from the joint distribution and a set
29  # of ovservations to condition on.
30  def probFluGivenObs(samples, obs):
31      # reject all samples which don't align
32      # with condition
33      keepSamples = []
34      for sample in samples:
35          if checkObsMatch(sample, obs):
36              keepSamples.append(sample)
37
38      # from remaining, simply count...
39      fluCount = 0
40      for sample in keepSamples:
41          [flu, und, fev, tir] = sample
42          if flu == 1:
43              fluCount += 1
44
45      # counting can be so sweet...
46      return float(fluCount) / len(keepSamples)
```

Alg #1: Rejection Sampling

```
 3 N_SAMPLES = 100000
 4
 5 # Program: Joint Sample
 6 # ----------------------
 7 # we can answer any pro
 8 # with multivariate sam
 9 # where conditioned var
10 def main():
11     obs = getObservatio
12     print 'Observation
13
14     samples = sampleATo
15     prob = probFluGiven
16     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, 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, 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!

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

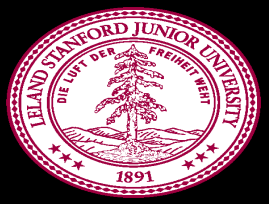
webMd — -bash — 39×20

```
[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
>
```

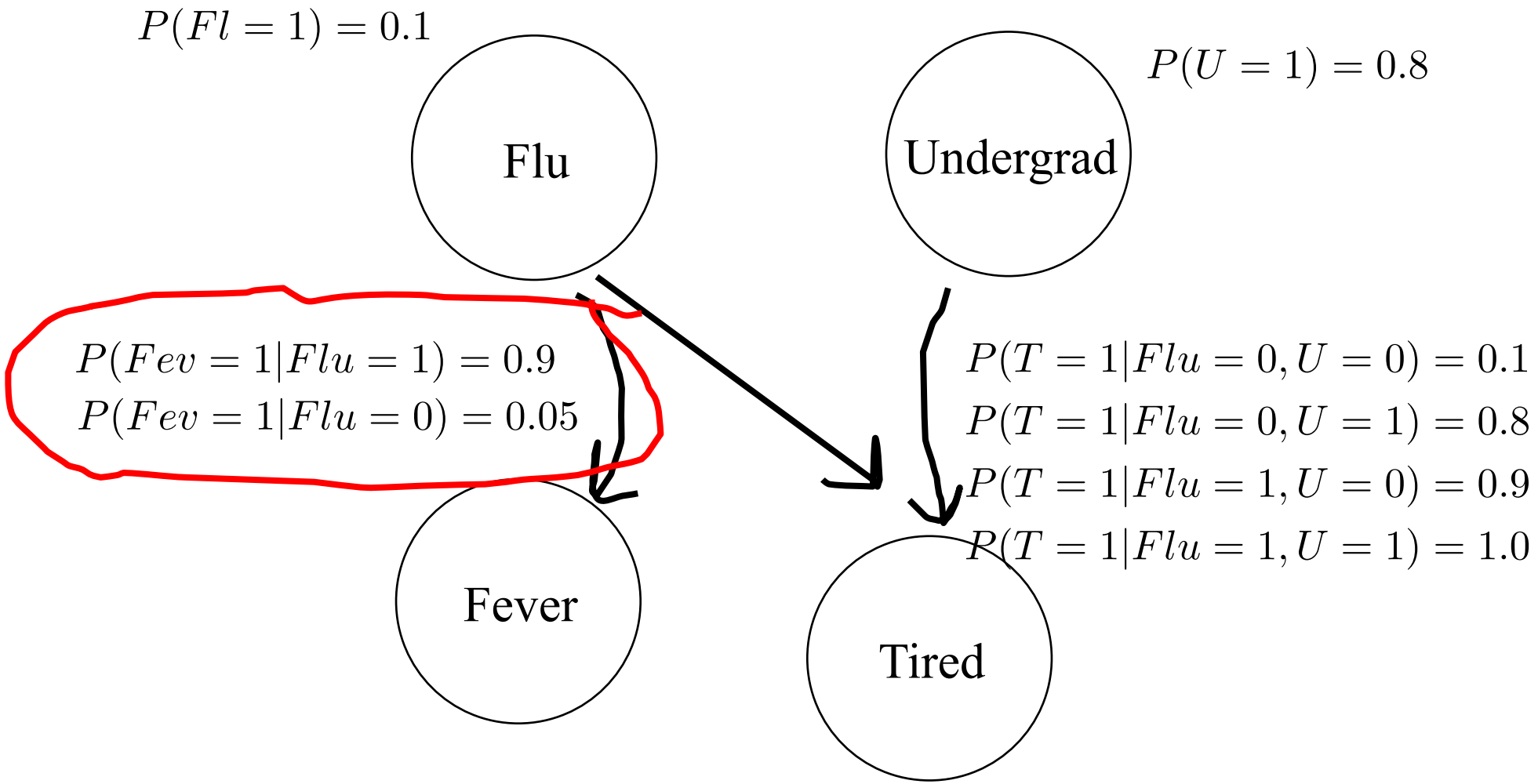
If you can sample enough from the joint distribution, you can answer any probability question

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

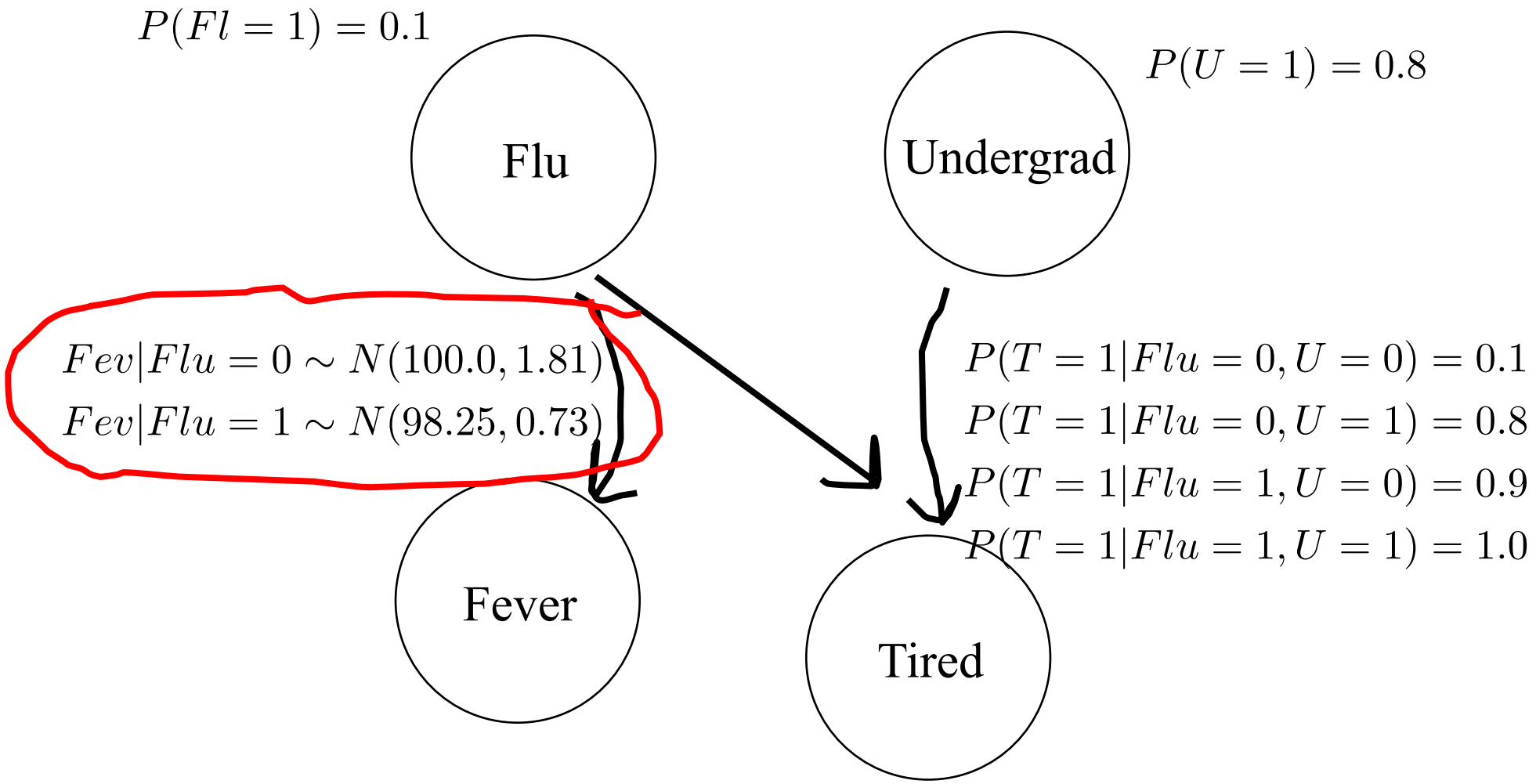
What's the matter with joint sampling?



Probabilistic Model



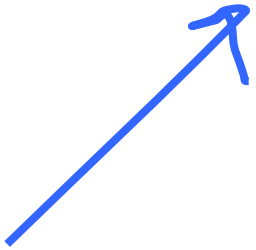
Probabilistic Model



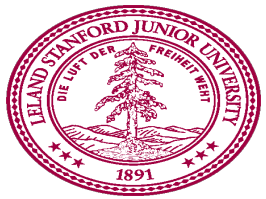
The Magic School Bus™

Markov Chain

MCMC



Monte Carlo



Alg #2: MCMC

```
webppl — -bash — 20×20
[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, 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]

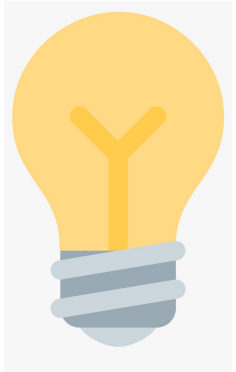
Many Algorithms

Rejection Sampling

MCMC

Pyro

Idea2Text



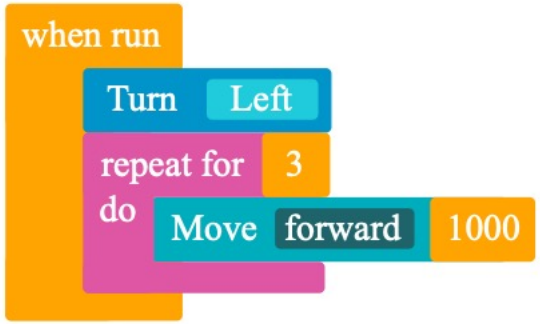
Stanford Acuity Test?
Version of rejection sampling

What about Code.org?

Computers Couldn't Understand Code

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

Challenge

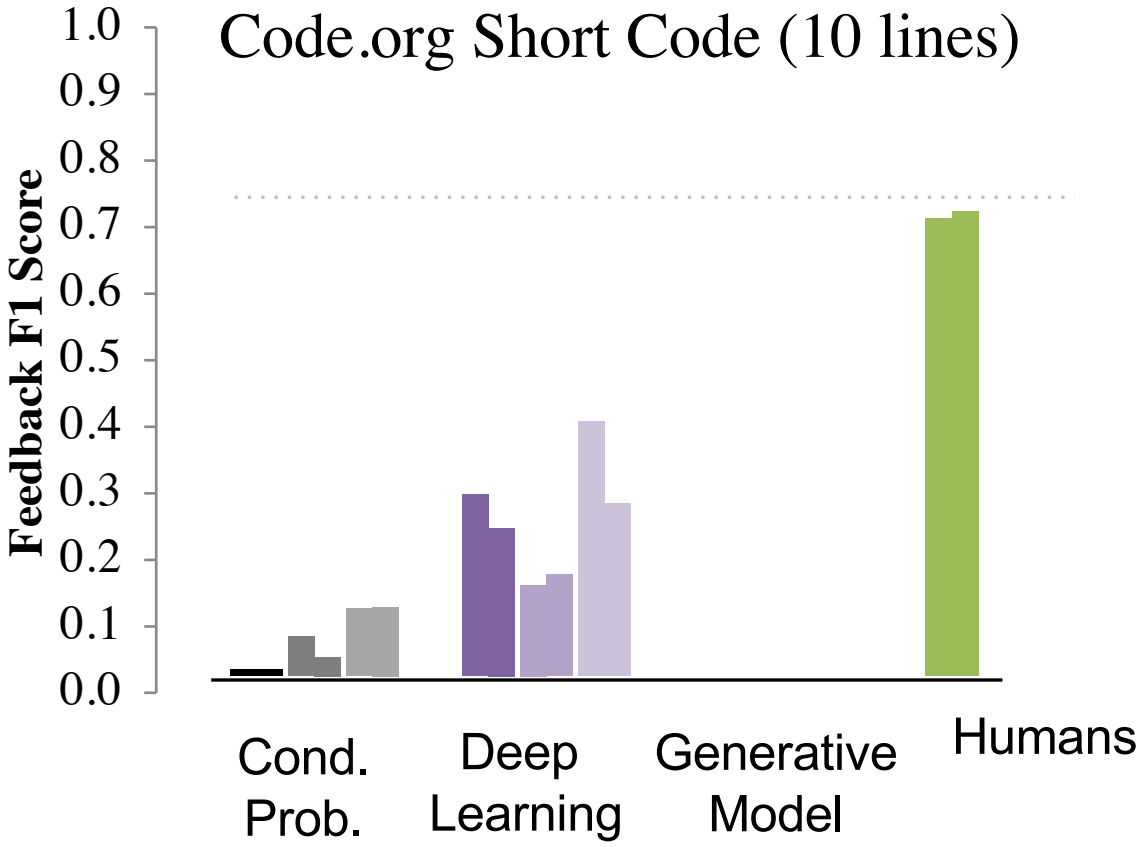


Student Code

You need to move and turn in your loop

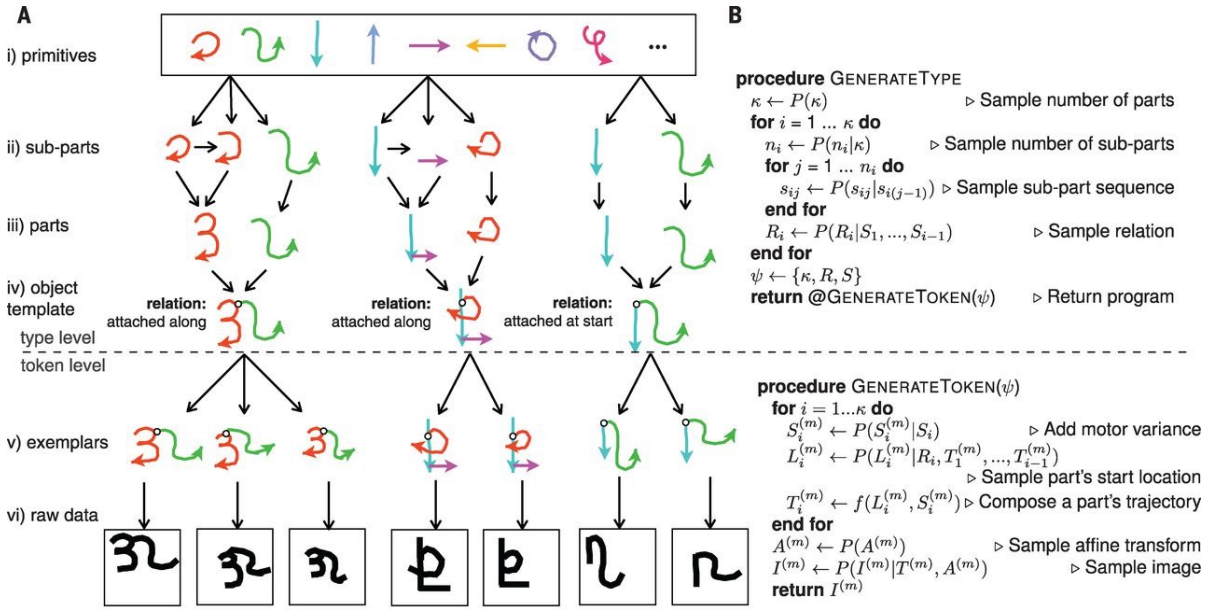
Insight

Computers Couldn't Understand Code

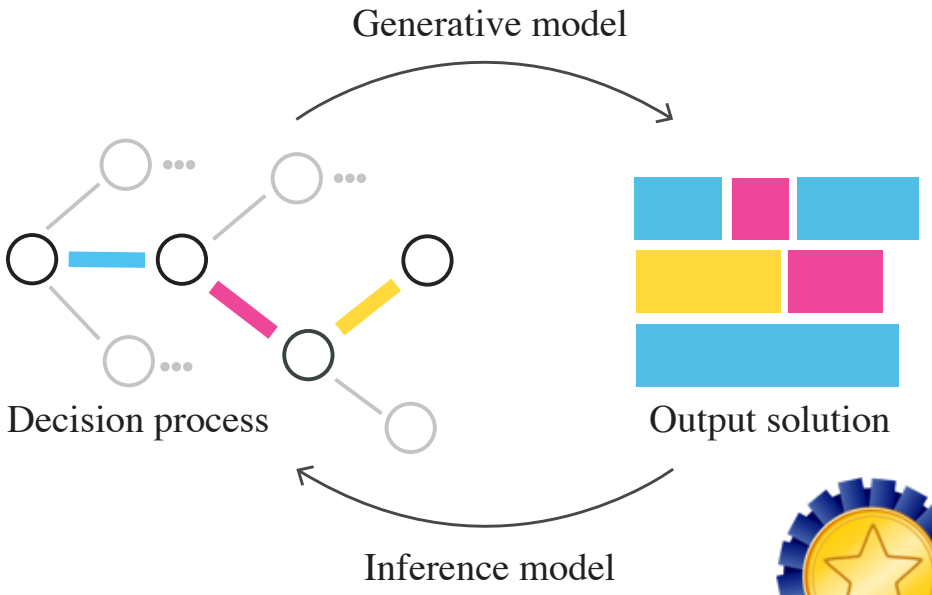


Generative Model of Grading

Lake et al, 2015

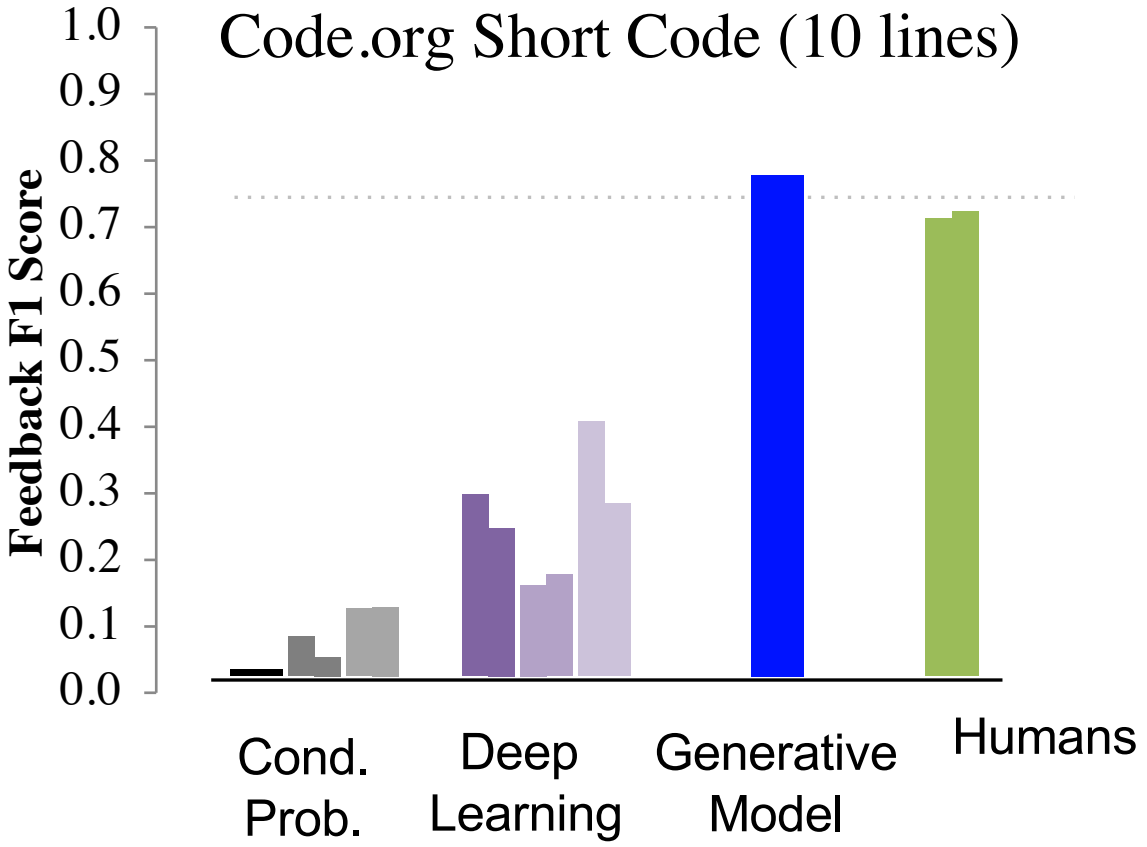


Muke Wu, Ali Malik, Noah Goodman, Chris Piech, 2019

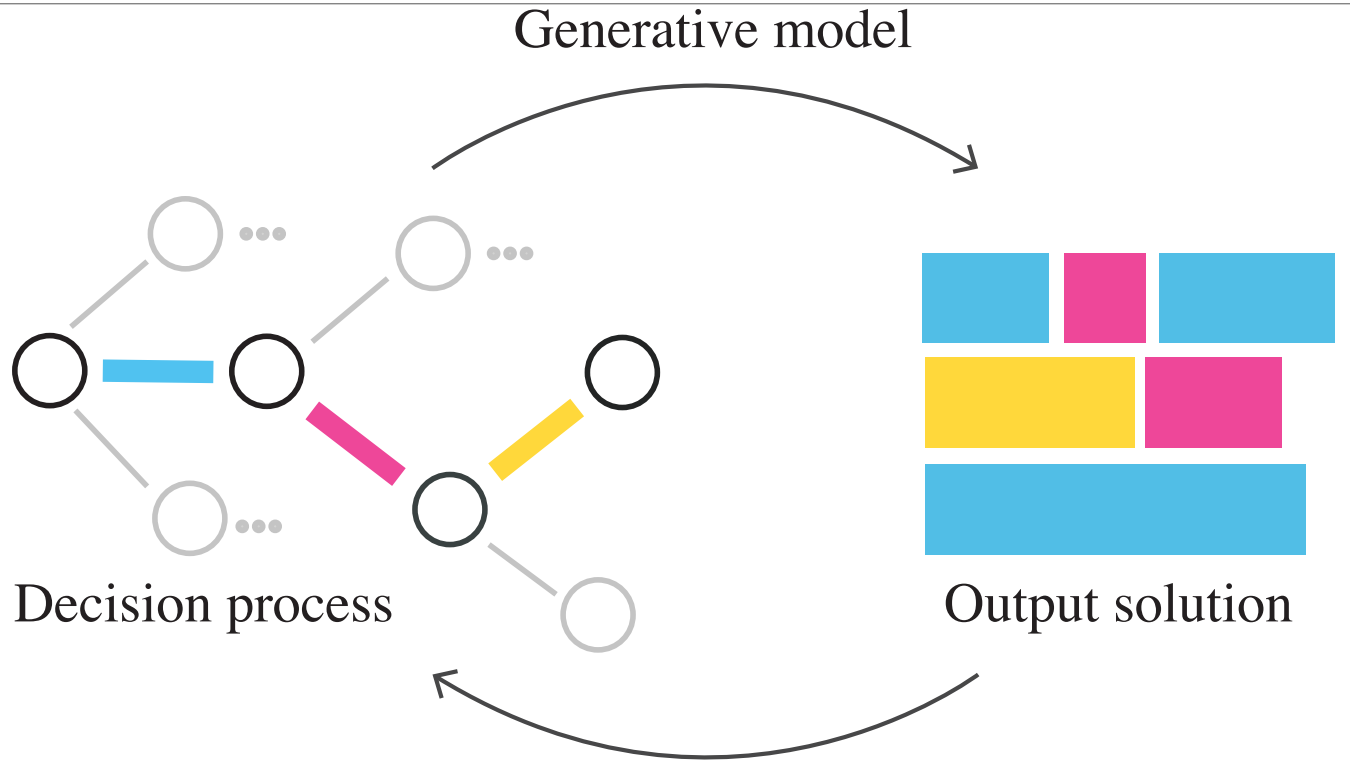


Outstanding Student paper award, AAAI 2019

Computers Couldn't Understand Code



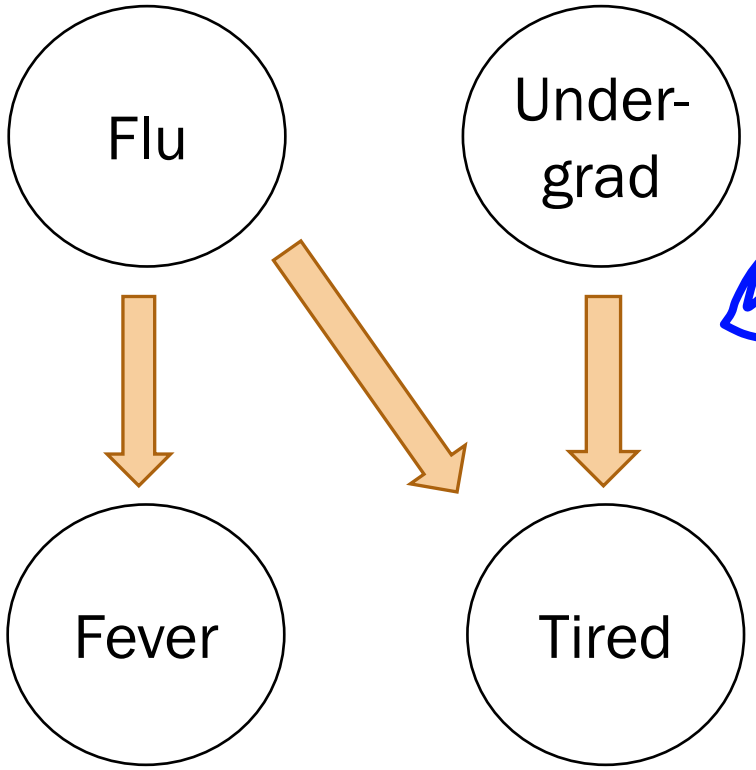
Idea 2 Text


Generative model
Decision process
Output solution
Inference model

What 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

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

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

Que te vayas bien