# 15: General Inference

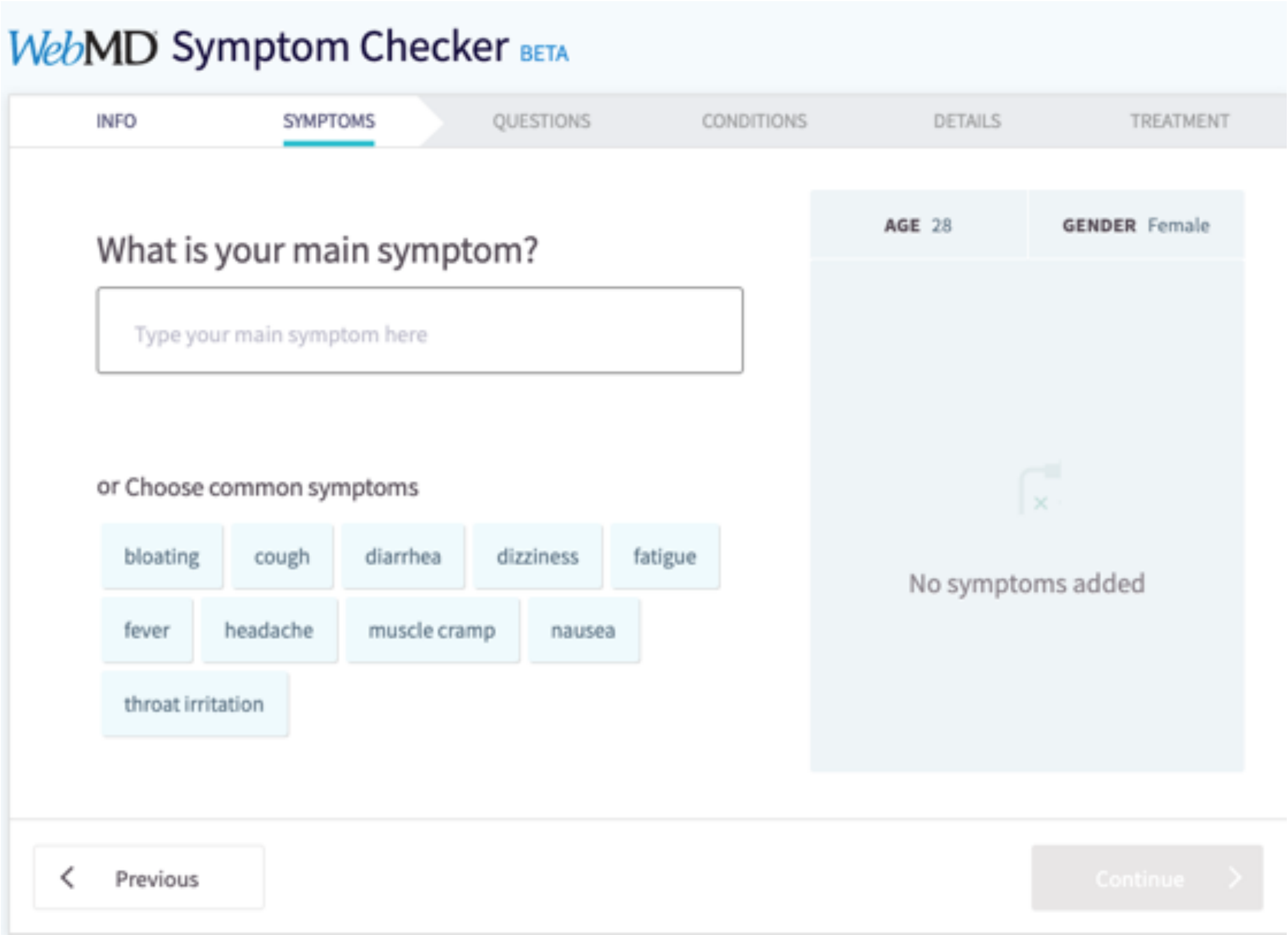
Lisa Yan and Jerry Cain
October 16, 2020

# Quick slide reference

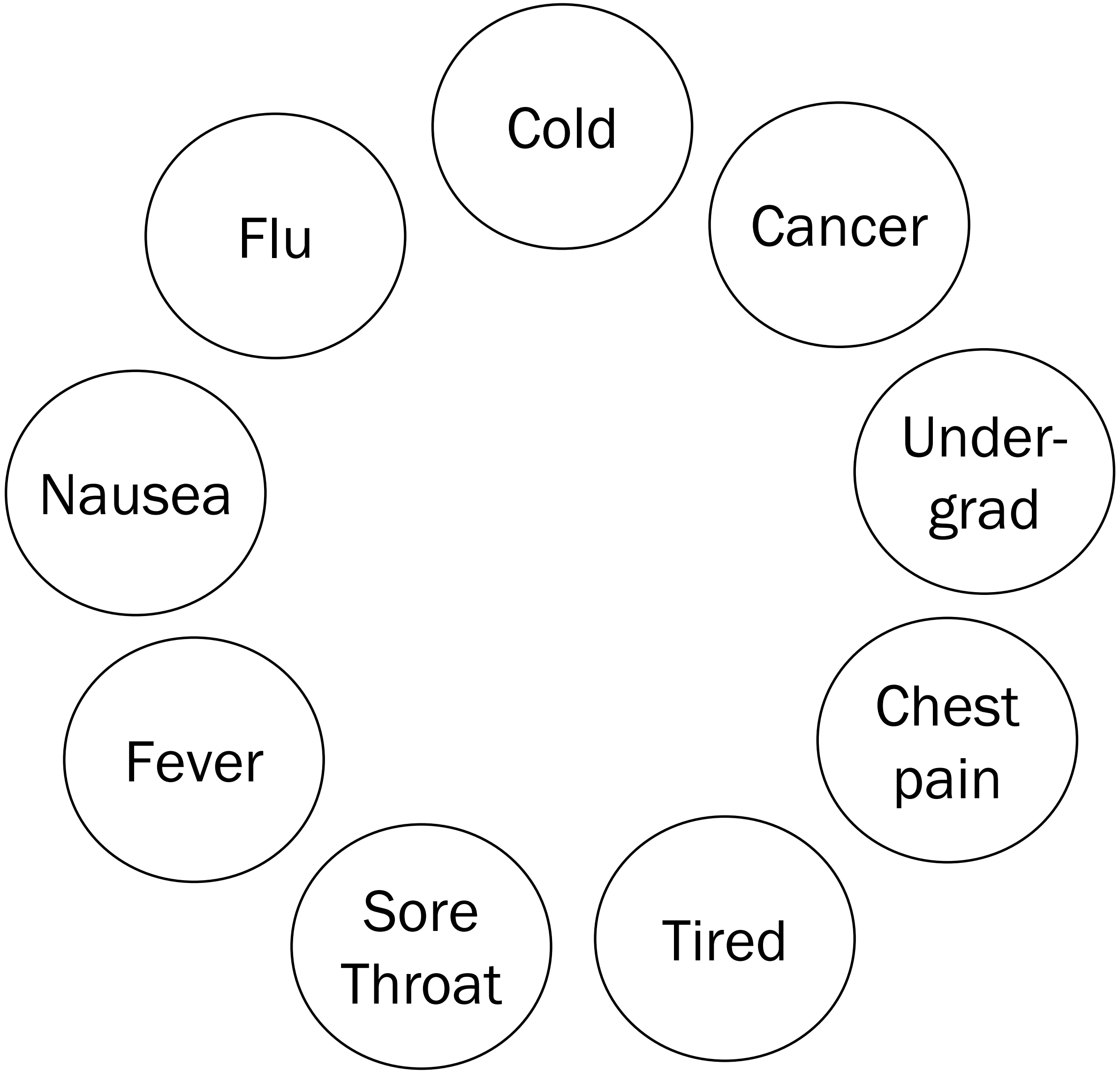
# General Inference: Introduction

# Inference

# Inference


WebMD Symptom Checker BETA
INFO
SYMPTOMS
QUESTIONS
CONDITIONS
DETAILS
TREATMENT
What is your main symptom?
Type your main symptom here
or Choose common symptoms
bloating
cough
diarrhea
dizziness
fatigue
fever
headache
muscle cramp
nausea
throat irritation
AGE 28
GENDER Female
No symptoms added
Previous
Continue

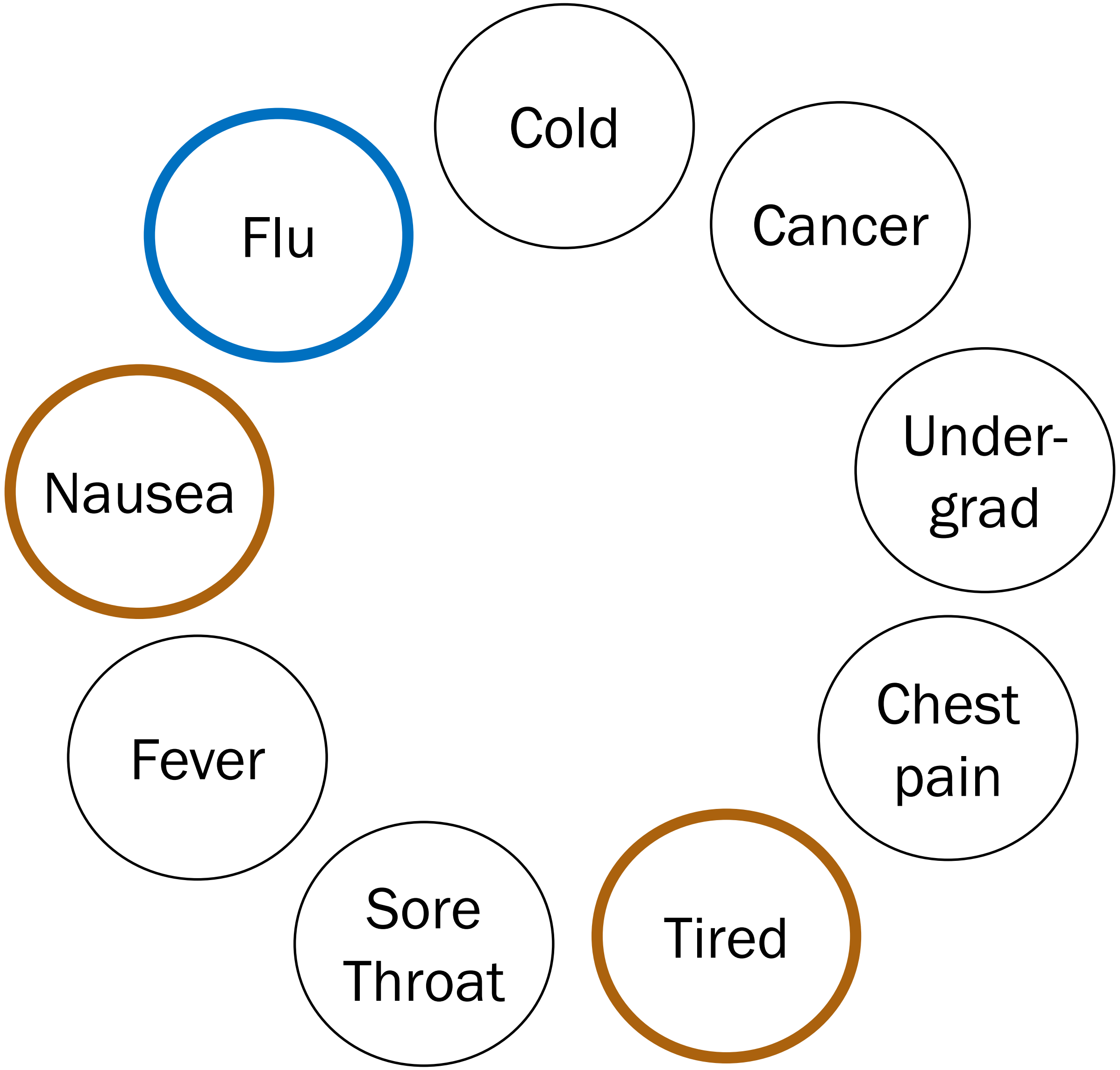
# Inference




General inference question:

Given the values of some random variables, what is the conditional distribution of some other random variables?

# Inference

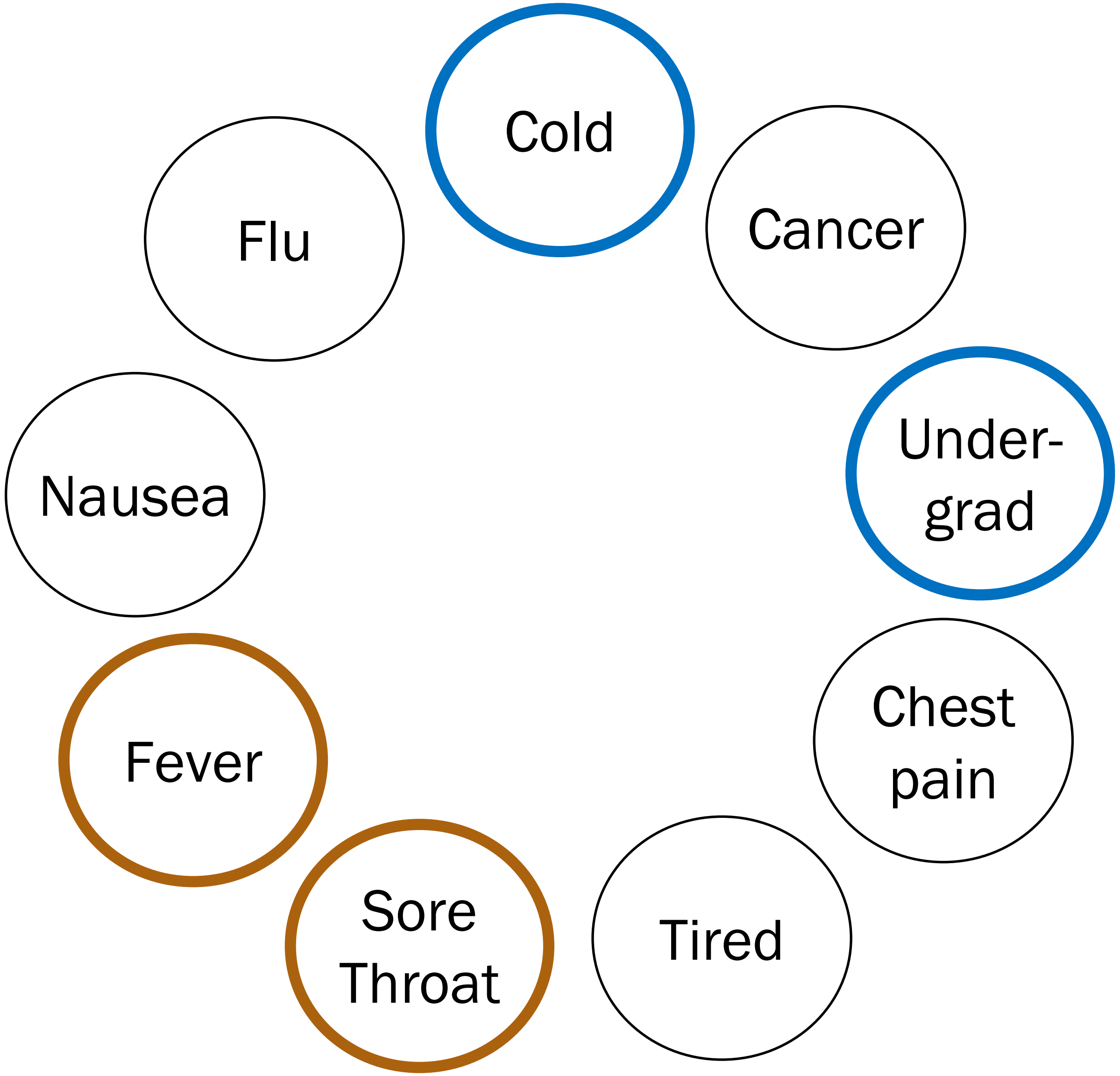



One inference question:

$$P(F = 1 | N = 1, T = 1)$$

$$= \frac{P(F = 1, N = 1, T = 1)}{P(N = 1, T = 1)}$$

# Inference

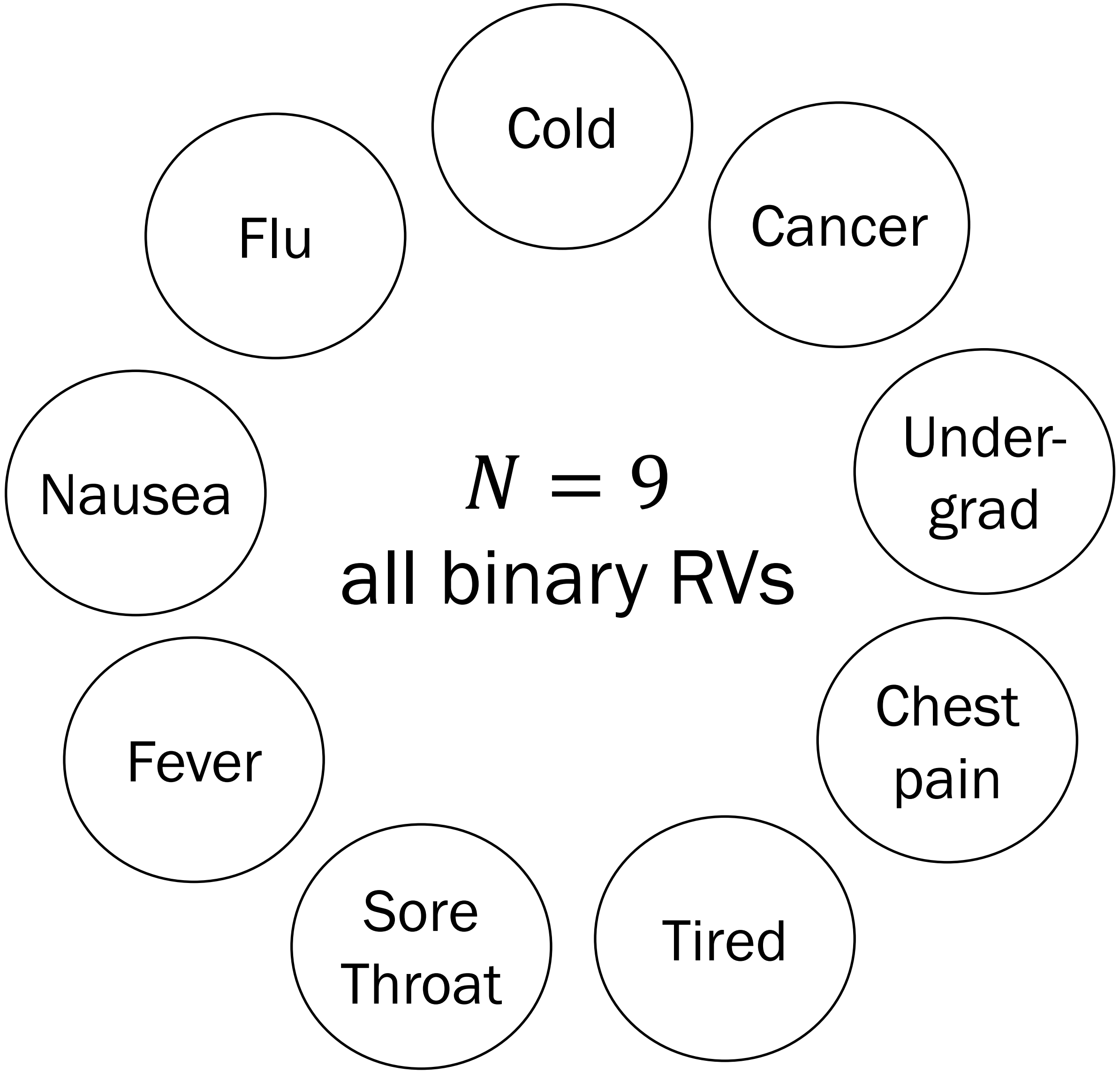



Another inference question:

$$P(C_o = 1, U = 1 | S = 0, F_e = 0)$$

$$= \frac{P(C_o = 1, U = 1, S = 0, F_e = 0)}{P(S = 0, F_e = 0)}$$

# Inference

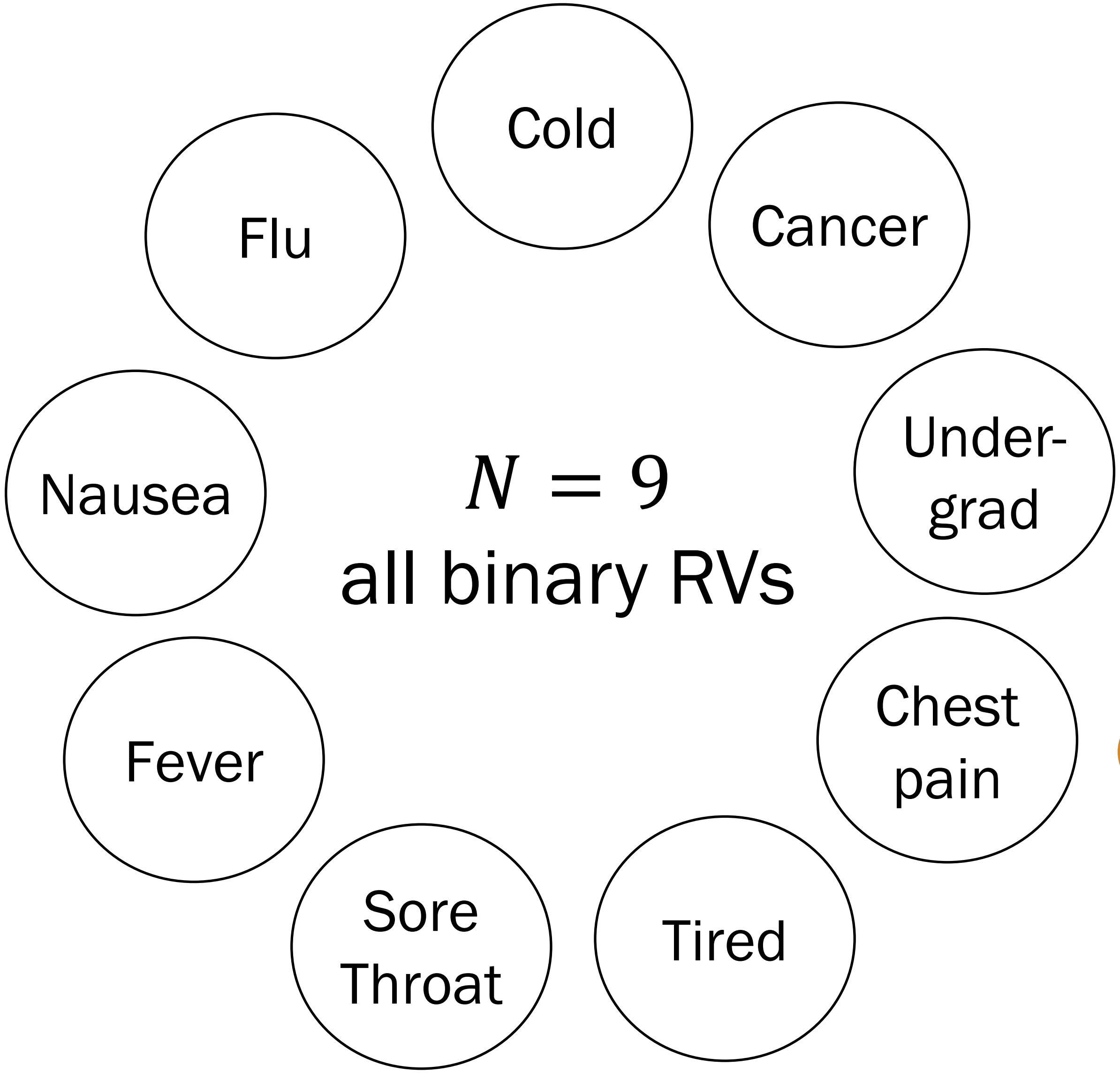



If we knew the **joint distribution**, we can answer all probabilistic inference questions.

What is the size of the joint probability table?

A. $2^{N-1}$ entries
B. $N^2$ entries
C. $2^N$ entries
D. None/other/don't know

# Inference



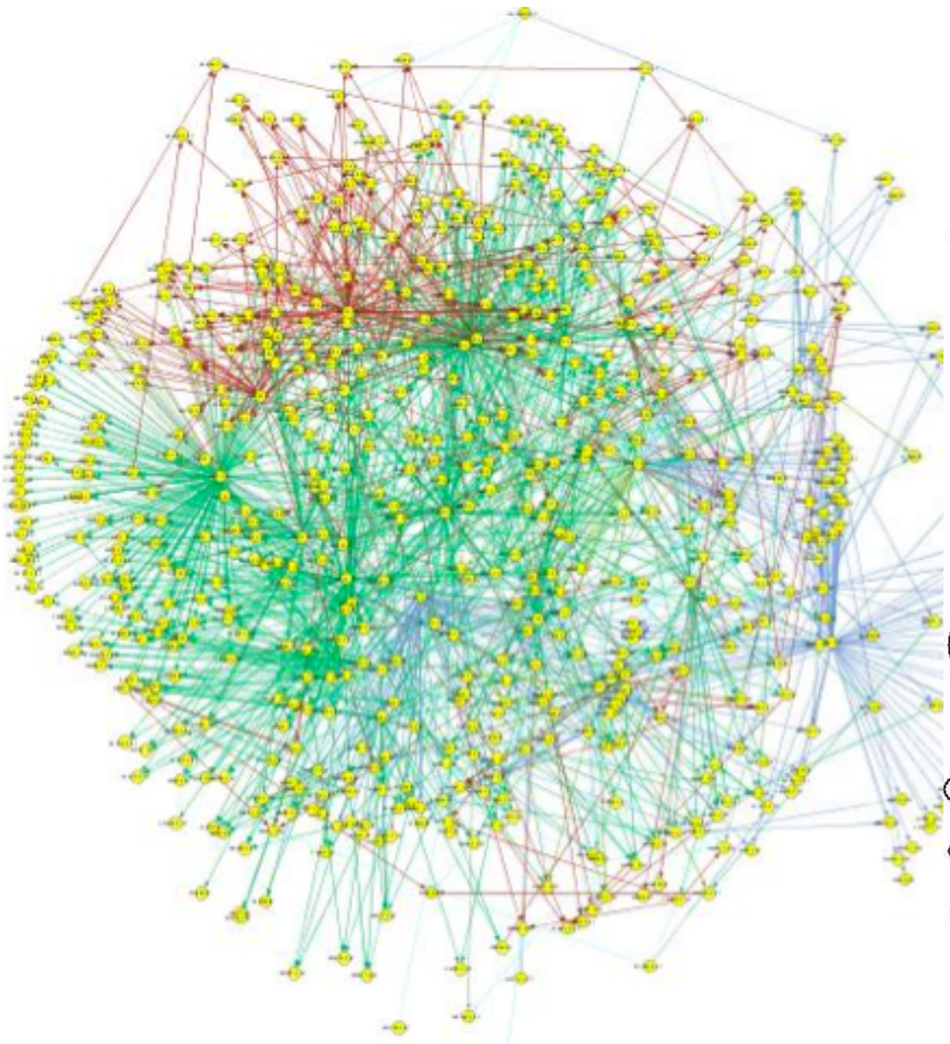

If we knew the **joint distribution**, we can answer all probabilistic inference questions.

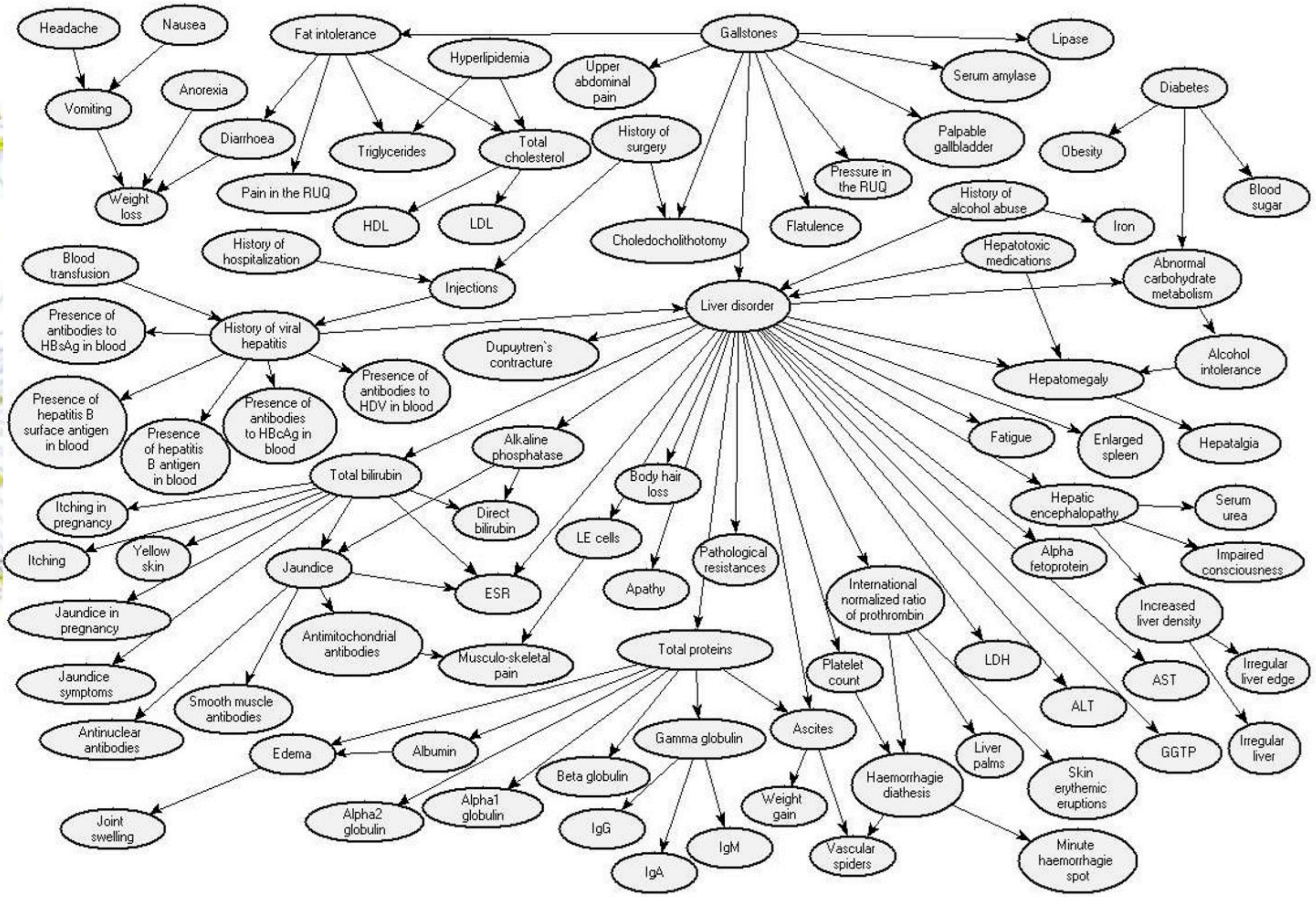
What is the size of the joint probability table?

A. $2^{N-1}$ entries
B. $N^2$ entries
C. $2^N$ entries
D. None/other/don't know

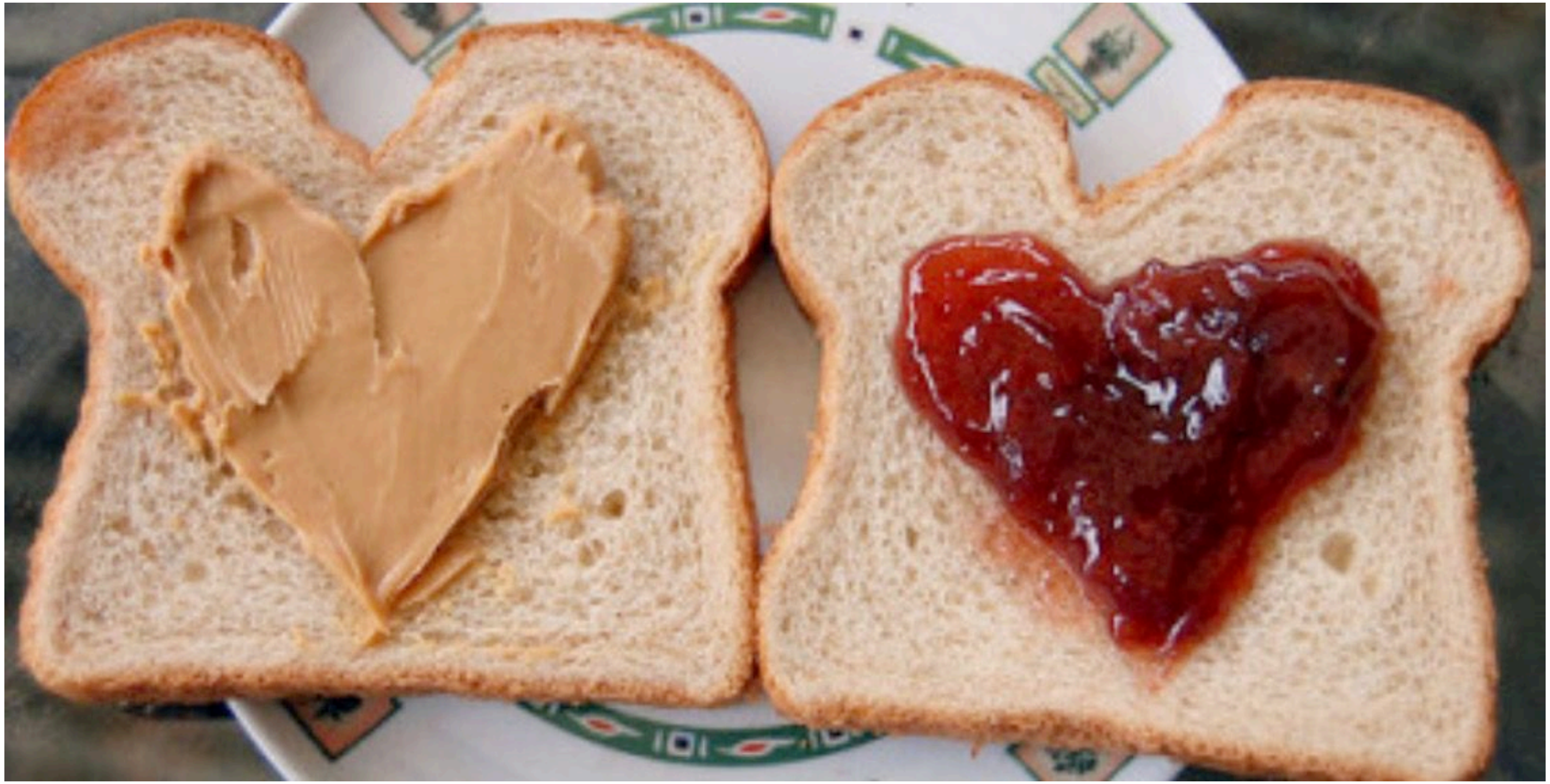
Naively specifying a joint distribution is often intractable.

# N can be large…





# Conditionally Independent RVs



~~Conditional Probability~~
Conditional Distributions

~~Independence~~
Independent RVs

# Conditionally Independent RVs

Recall that two events $A$ and $B$ are conditionally independent given $E$ if:

$$P(AB|E) = P(A|E)P(B|E)$$

$n$ discrete random variables $X_1, X_2, \ldots, X_n$ are called **conditionally independent given $Y$** if:

for all $x_1, x_2, \ldots, x_n, y$:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n | Y = y) = \prod_{i=1}^{n} P(X_i = x_i | Y = y)$$

This implies the following (cool to remember for later):

$$\log P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n | Y = y) = \sum_{i=1}^{n} \log P(X_i = x_i | Y = y)$$

# Lec. 12: Independence of multiple random variables

Errata

Recall independence of $n$ events $E_1, E_2, \ldots, E_n$:

for $r = 1, \ldots, n$:

for every subset $E_1, E_2, \ldots, E_r$:

$$P(E_1, E_2, \ldots, E_r) = P(E_1)P(E_2)\cdots P(E_r)$$

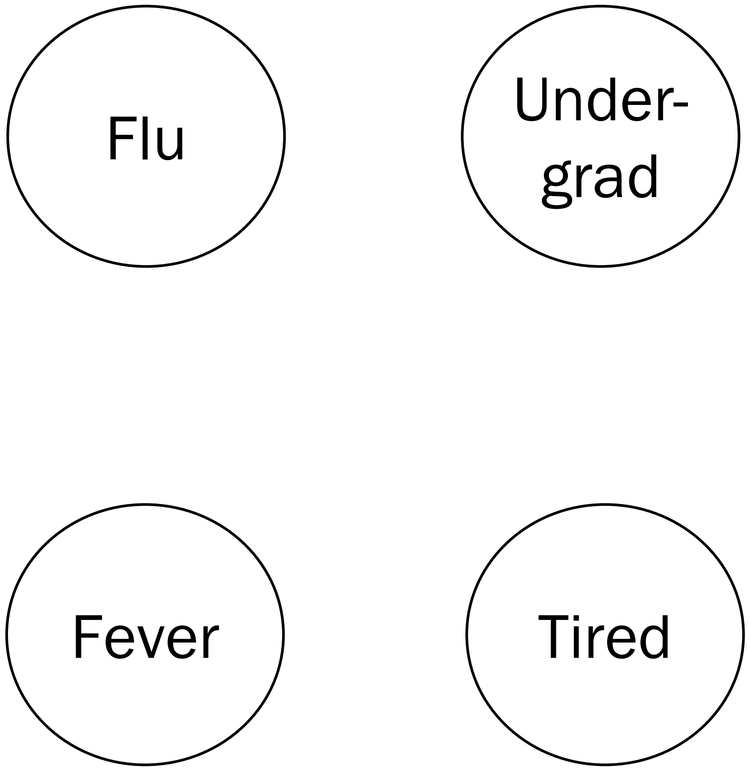
We have independence of $n$ discrete random variables $X_1, X_2, \ldots, X_n$ if for all $x_1, x_2, \ldots, x_n$:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i)$$

Errata (edited May 3): **Removed the independent RV requirement for all subsets of size $r = 1, \ldots, n$.** Do you see why this requirement is unnecessary?
(Hint: independence of RVs implies independence of all events)

# Bayesian Networks

# A simpler WebMD



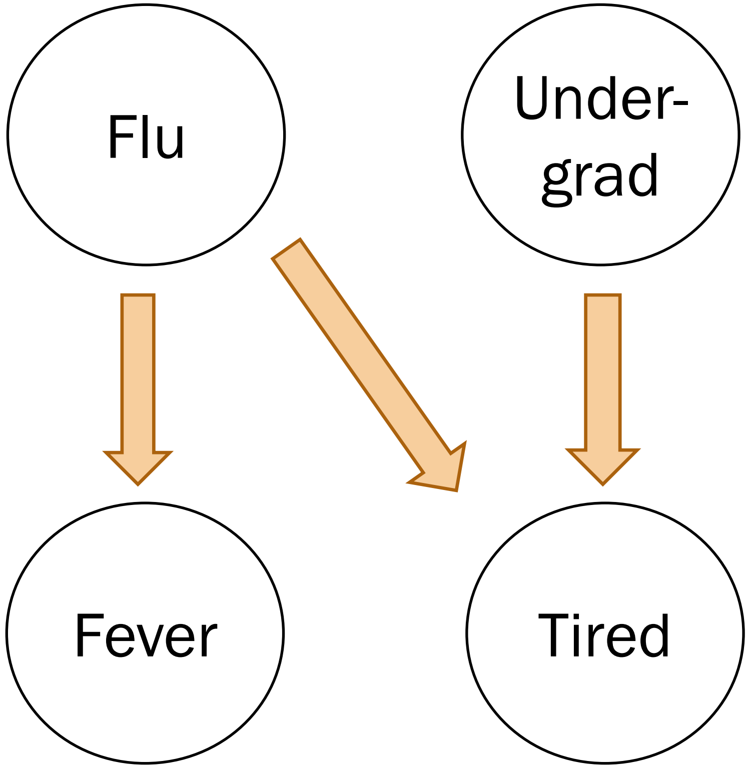

Great! Just specify $2^4 = 16$ joint probabilities...?

$$P(F_{lu} = a, F_{ev} = b, U = c, T = d)$$

What would a Stanford flu expert do?

Describe the joint distribution using causality!!!

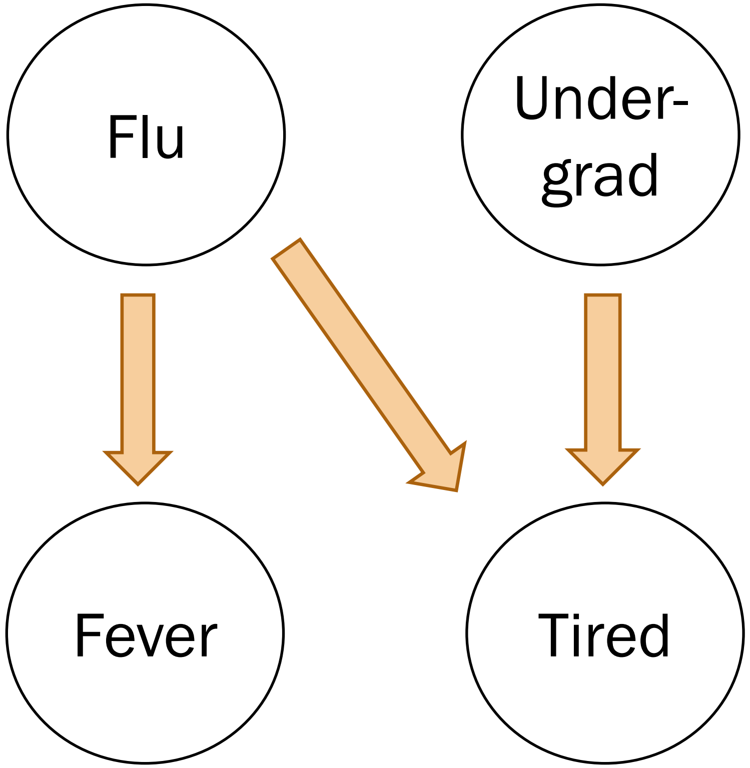
# Constructing a Bayesian Network




What would a Stanford flu expert do?

1. Describe the joint distribution using causality.
2. **Assume conditional independence.**

# Constructing a Bayesian Network




In a Bayesian Network,

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

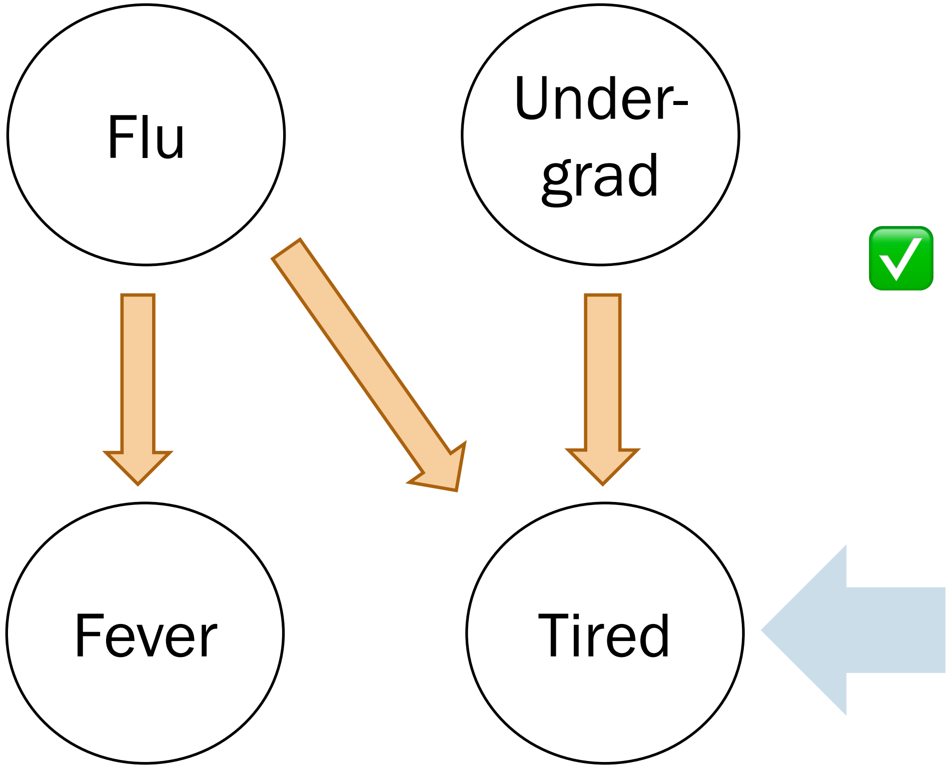
- Node: random variable
- Directed edge: conditional dependency

Examples:

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

# Constructing a Bayesian Network

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




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

What would a Stanford flu expert do?

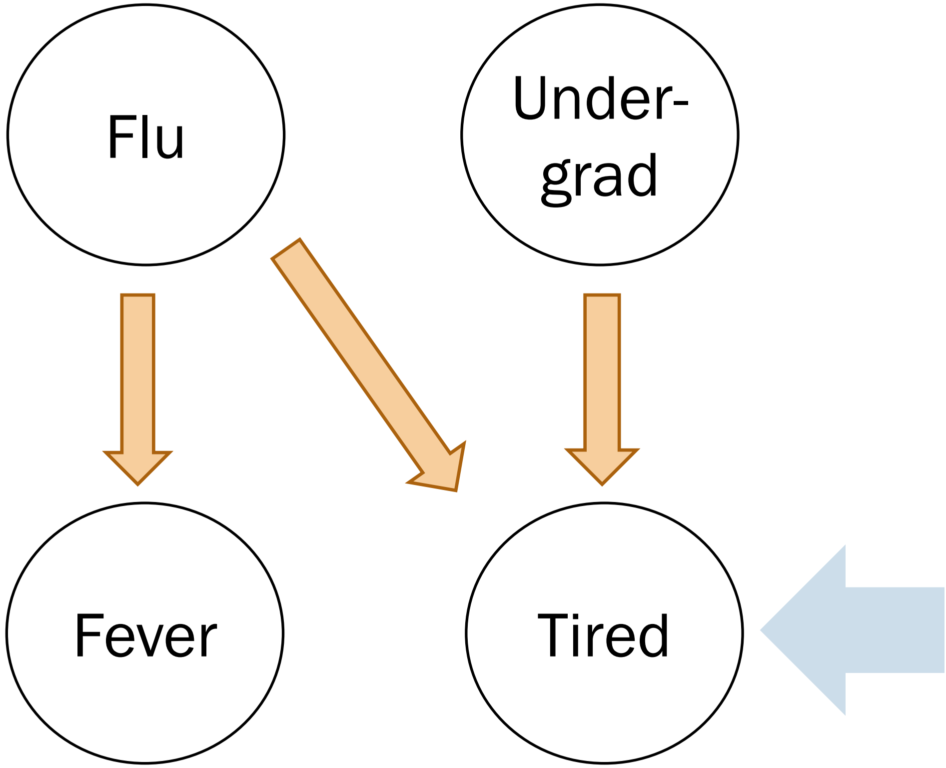
1. Describe the joint distribution using causality.
2. ✅ Assume conditional independence.
3. Provide $P$(values|parents) for each random variable

What conditional probabilities should our expert specify?

🤔

# Constructing a Bayesian Network

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




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

What would a Stanford flu expert do?

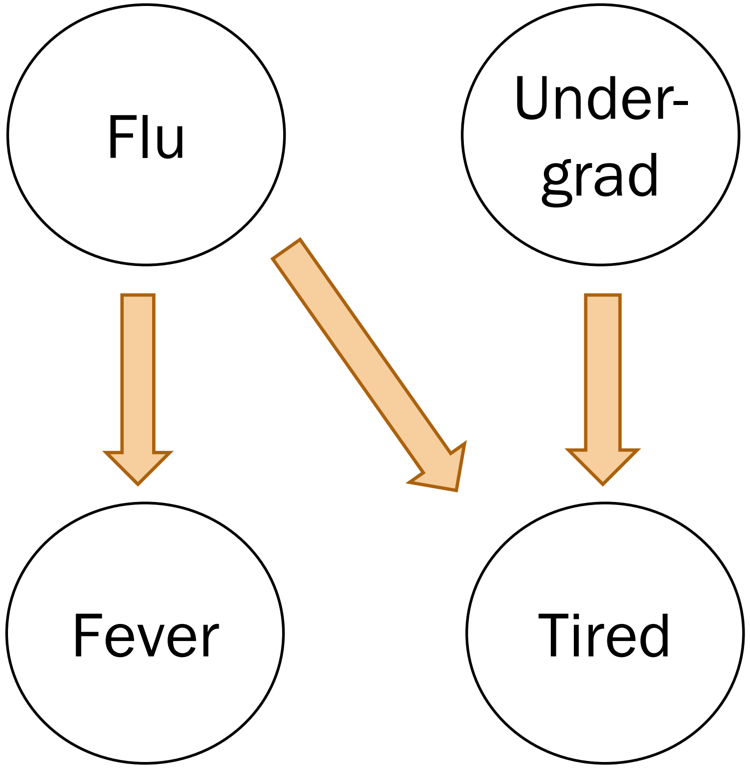
1. Describe the joint distribution using causality.
2. Assume conditional independence.
3. Provide $P$(values|parents) for each random variable

What conditional probabilities should our expert specify?

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

# Using a Bayes Net

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




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

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

What would a CS109 student do?

1. Populate a Bayesian network by asking a Stanford flu expert
   or
   by using reasonable assumptions

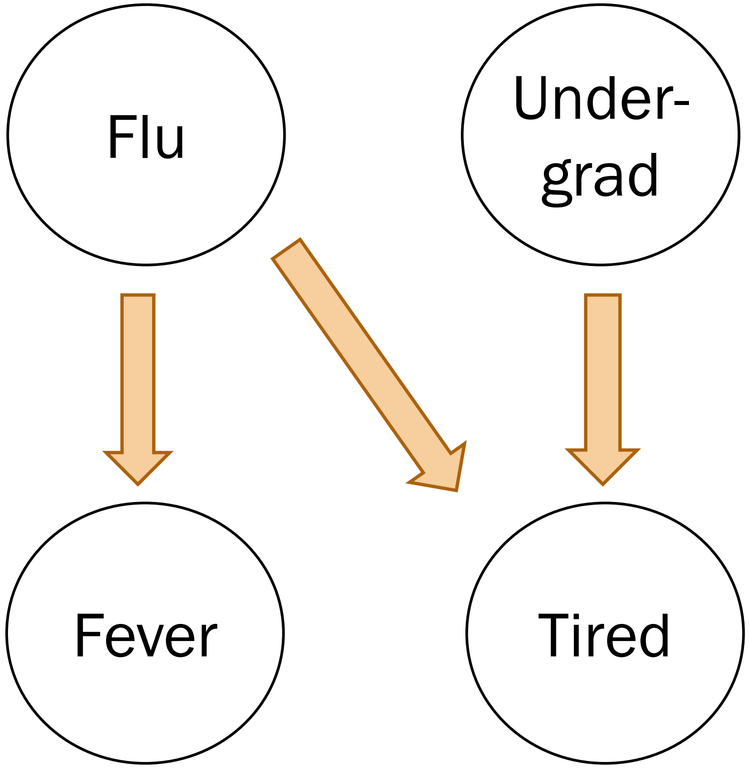
2. **Answer inference questions**

Our focus today

# Inference (I): Math

# Bayes Nets: Conditional independence





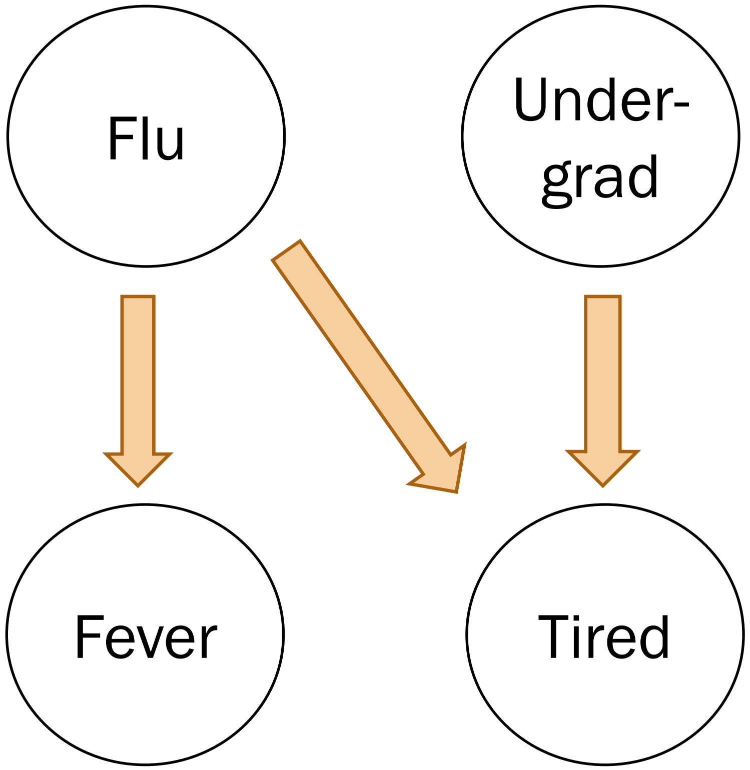

In a Bayesian Network,

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

- Node: random variable
- Directed edge: conditional dependency

# Inference via math

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




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

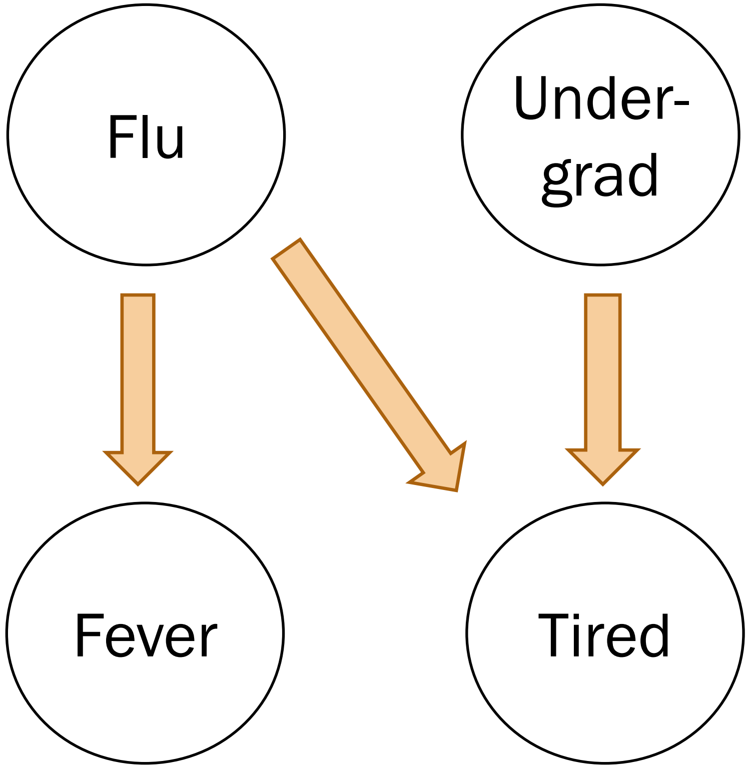
$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

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

   Compute joint probabilities using chain rule.

# Inference via math

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




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

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

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

   1. Compute joint probabilities

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

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

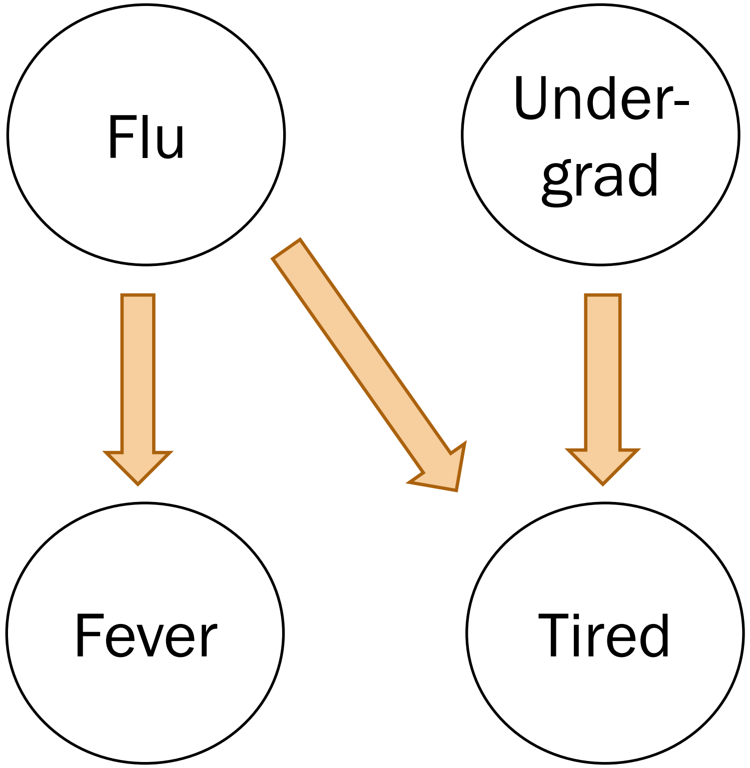
   2. Definition of conditional probability

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

$$= 0.095$$

# Inference via math

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



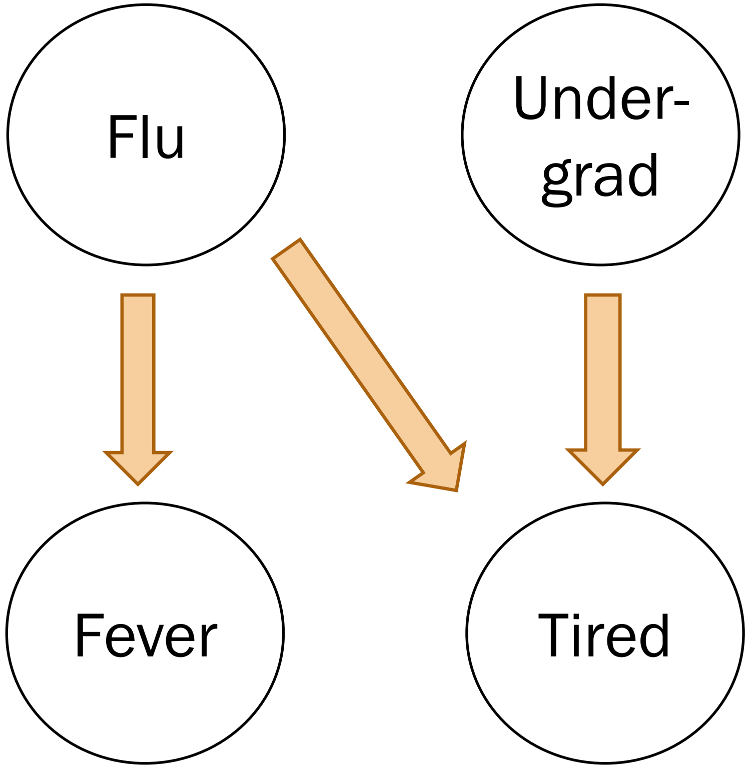

$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

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

# Inference via math

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




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

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

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

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

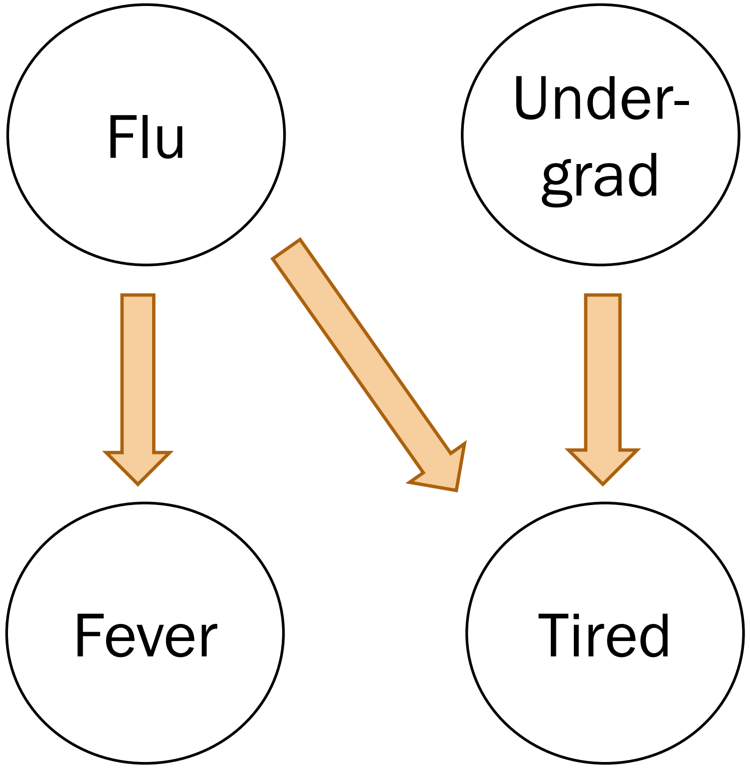
   2. Definition of conditional probability

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

$$= 0.122$$

# Inference via math

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




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

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

Solving inference questions precisely is possible, but sometimes tedious.

Can we use sampling to do approximate inference?

Yes.

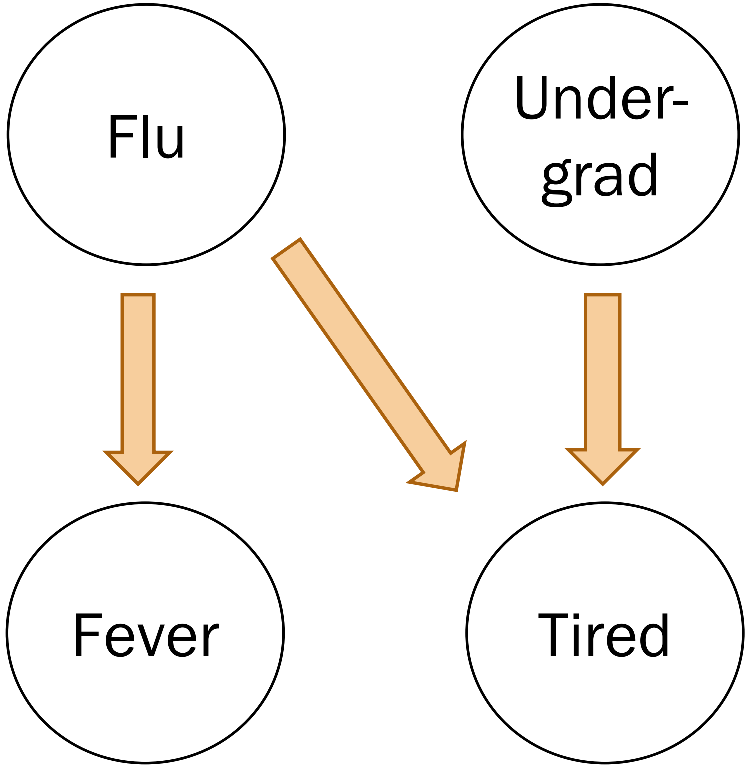
# (live)
# 15: General Inference

Lisa Yan and Jerry Cain

October 16, 2020

# Constructing a Bayesian Network






In a Bayesian Network,

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

- Node: random variable
- Directed edge: conditional dependency

Examples:

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

# Breakout Rooms

Check out the question on the next slide (Slide 32). Post any clarifications here!

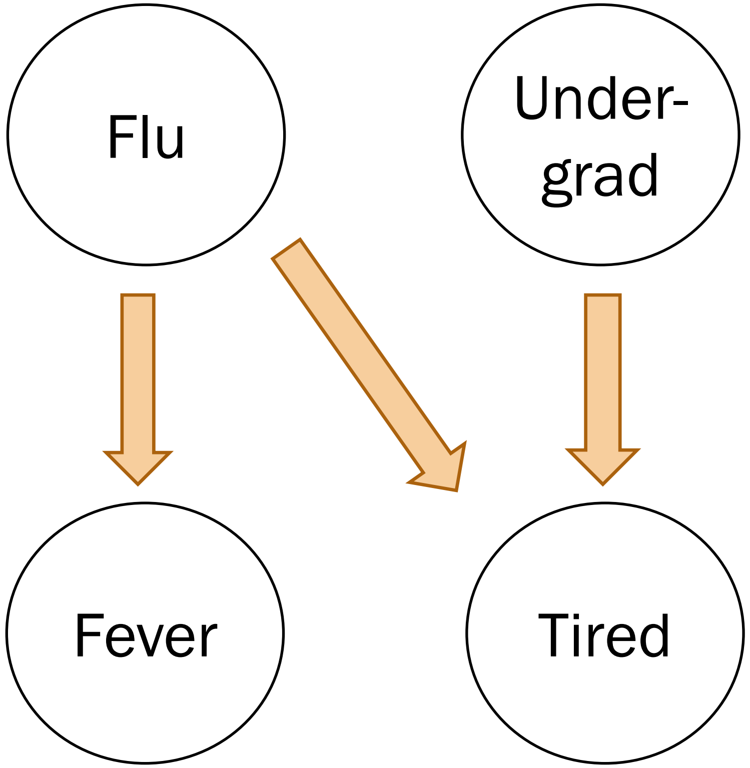
https://us.edstem.org/courses/2678/discussion/146234

Breakout rooms: 4 min. Introduce yourself!

# Inference via math

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

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



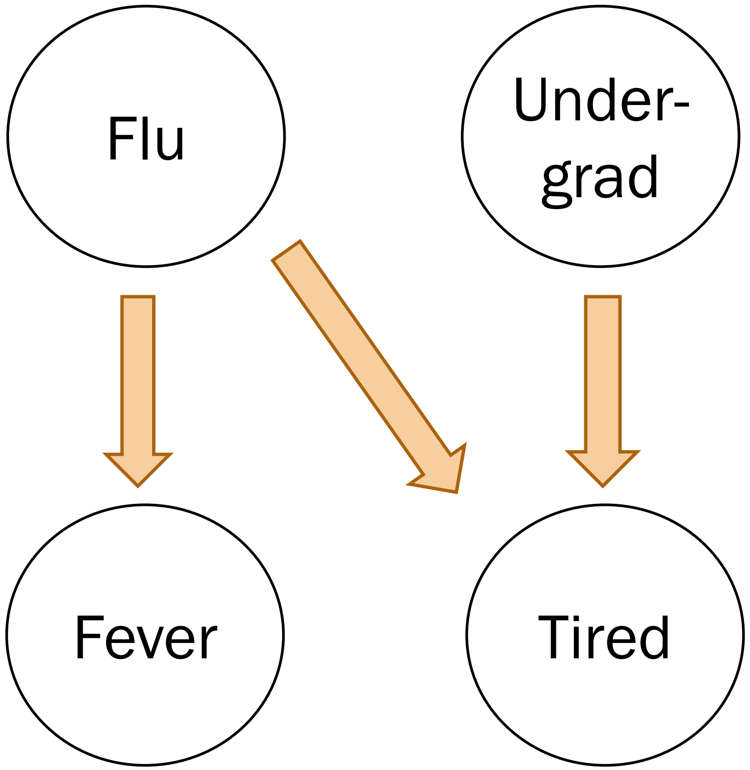

$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

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

# Inference via math

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

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

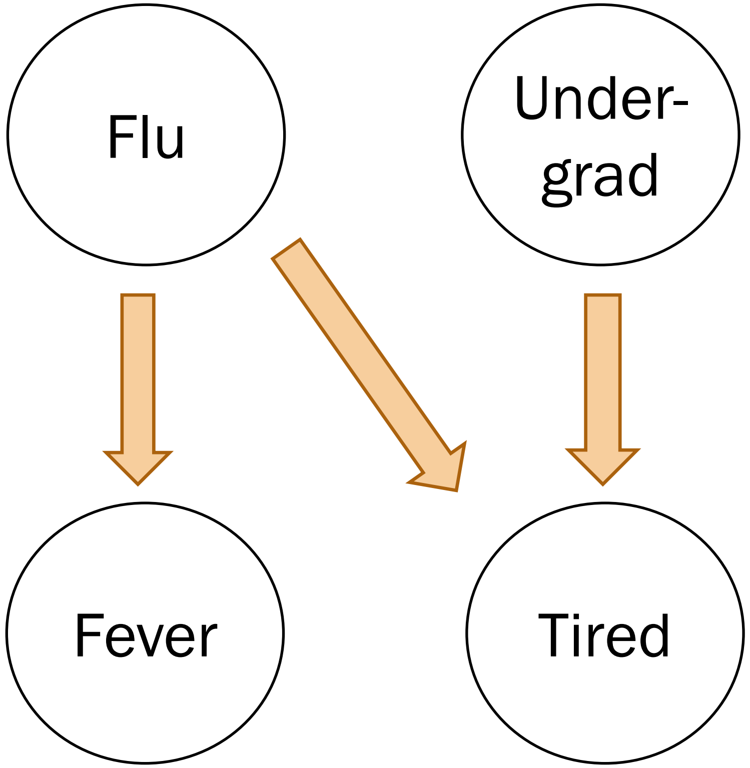



$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

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

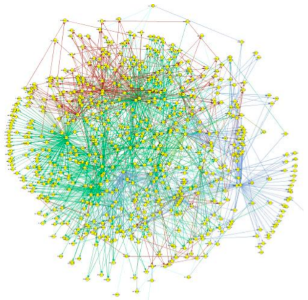
# Inference via math

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




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

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



Solving inference questions precisely is possible, but sometimes tedious.

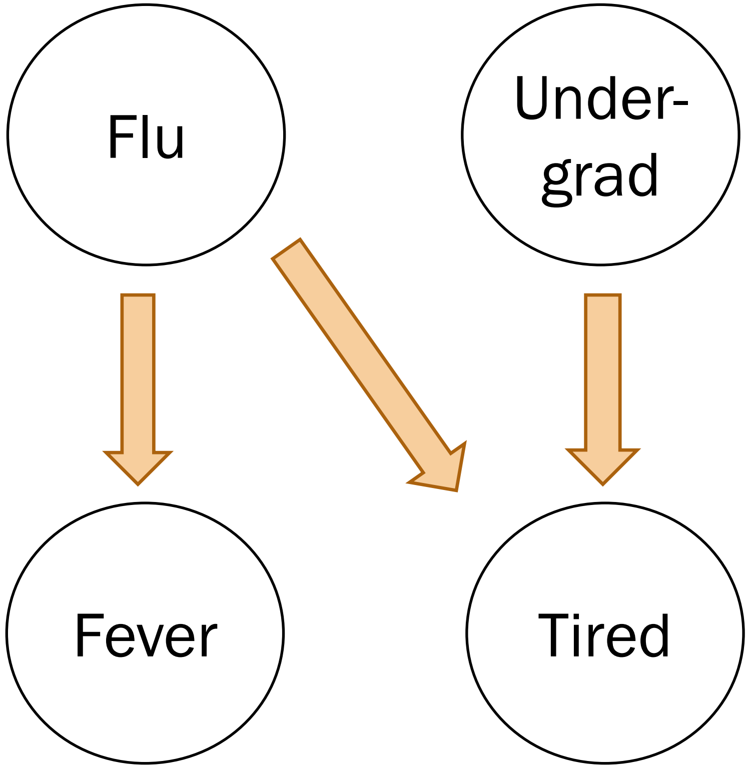
Can we use sampling to do approximate inference?

Yes.

# Inference via math



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




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

$= 0.122$

(from pre-lecture video)

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

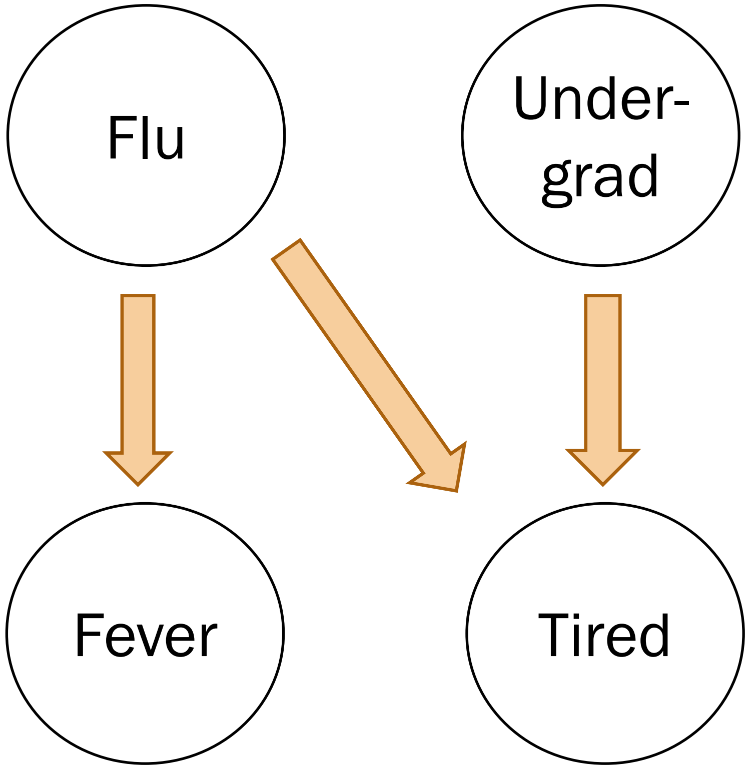
$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

# Rejection sampling algorithm

Step 0:

Have a fully specified Bayesian Network

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




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

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

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
    samples = sample_a_ton()
    samples_observation = …
          # number of samples with (U = 1, T = 1)
    samples_event =
          # number of samples with (F_lu = 1, U = 1, T
    return len(samples_event)/len(samples_obser
```

[flu, und, fev, tir]

```
Sampling...
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 1, 1]
[0, 1, 0, 0]
[1, 1, 1, 1]
[0, 0, 1, 1]
...
[0, 1, 0, 1]
Finished sampling
```

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation = …
        # number of samples with (U = 1, T = 1)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T = 1)
  return len(samples_event)/len(samples_observation)
```

Approximate Probability = $\dfrac{\#\text{ samples with } (F_{lu} = 1, U = 1, T = 1)}{\#\text{ samples with } (U = 1, T = 1)}$

# Rejection sampling algorithm

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

Approximate Probability = $\dfrac{\#\text{ samples with }(F_{lu} = 1, U = 1, T = 1)}{\#\text{ samples with }(U = 1, T = 1)}$

Why would this definition of approximate probability make sense?

# Think

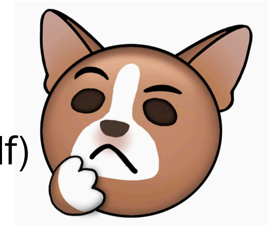
Slide 41 has a question to go over by yourself.

Post any clarifications here!

https://us.edstem.org/courses/2678/discussion/146234

Think by yourself: 2 min

(by yourself)



# Why would this approximate probability make sense?

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

Approximate Probability = $\dfrac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$

Recall our definition of probability as a frequency: $P(E) = \lim_{n \to \infty} \dfrac{n(E)}{n}$

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

# Why would this approximate probability make sense?

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

Approximate Probability = $\dfrac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$

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

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation = …
        # number of samples with (U = 1, T = 1)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T
  return len(samples_event)/len(samples_obser
```

[flu, und, fev, tir]

```
Sampling...
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 1, 1]
[0, 1, 0, 0]
[1, 1, 1, 1]
[0, 0, 1, 1]
...
[0, 1, 0, 1]
Finished sampling
```

# Rejection sampling algorithm

```
N_SAMPLES = 100000
# Method: Sample a ton
# -------------------
# create N_SAMPLES with likelihood proportional
# to the joint distribution
def sample_a_ton():
    samples = []
    for i in range(N_SAMPLES):
        sample = make_sample() # a particle
        samples.append(sample)
    return samples
```

How do we make a sample $(F_{lu} = a, U = b, F_{ev} = c, T = d)$ according to the joint probability?

Create a sample using the Bayesian Network!!

# Rejection sampling algorithm

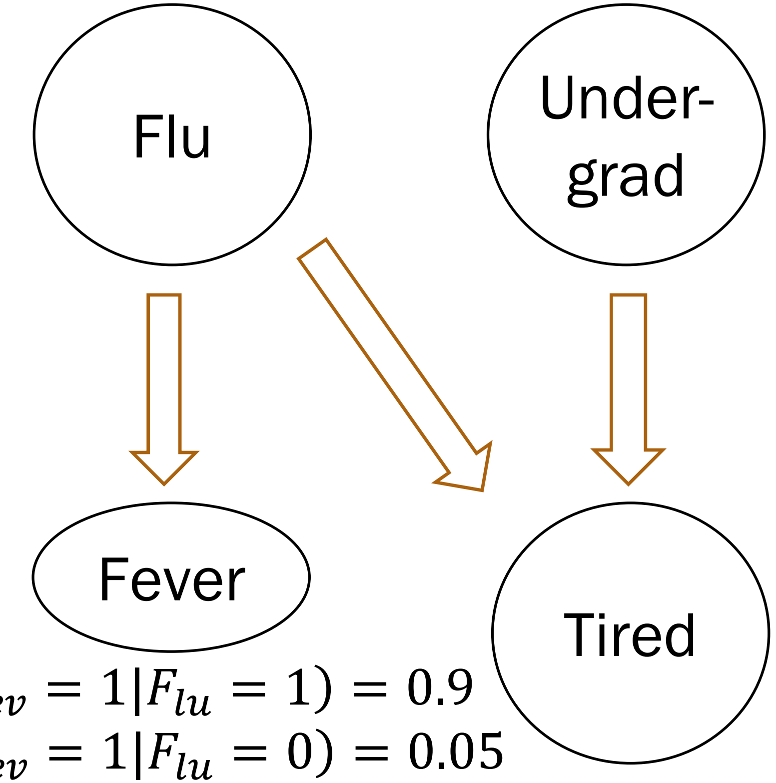
```
# Method: Make Sample
# -------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

    # choose tired based on (undergrad and flu)
    #
    # TODO: fill in
    #
    #

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

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




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

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

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

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

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

$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

# Rejection sampling algorithm

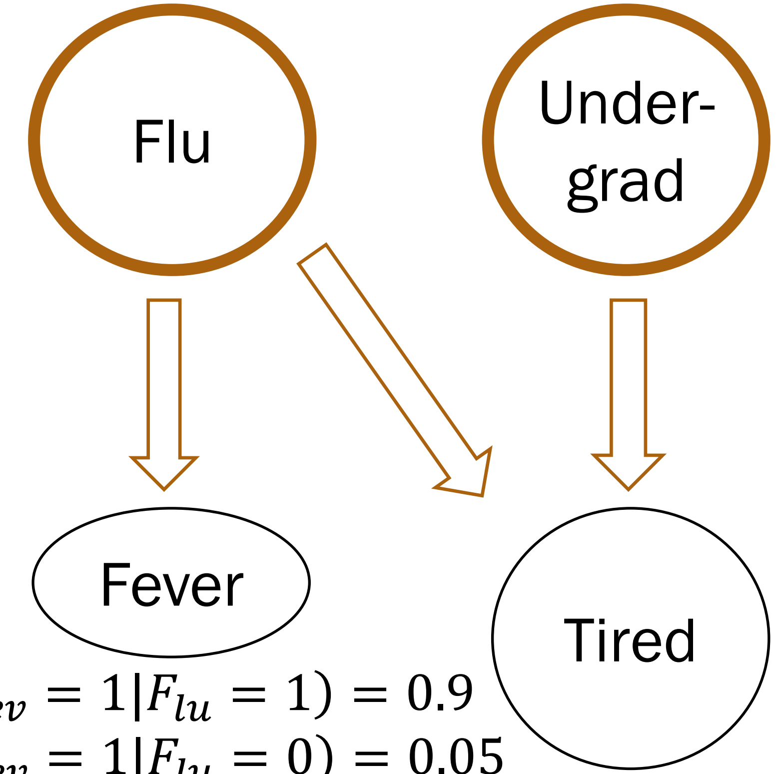
```
# Method: Make Sample
# ---------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

    # choose tired based on (undergrad and flu)
    #
    # TODO: fill in
    #
    #

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

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




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

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

# Rejection sampling algorithm

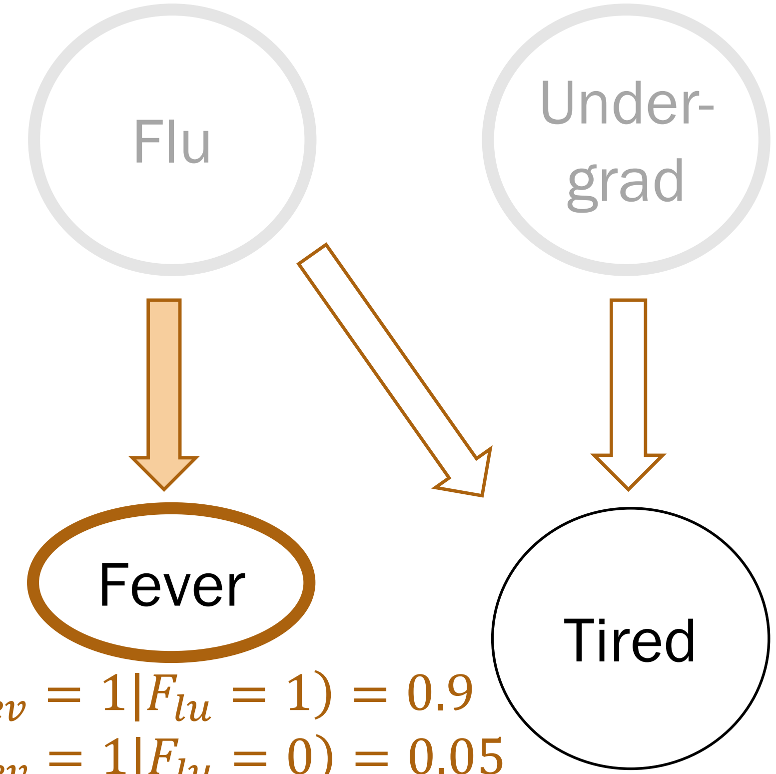
```
# Method: Make Sample
# --------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

    # choose tired based on (undergrad and flu)
    #
    # TODO: fill in
    #
    #

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

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




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

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

# Rejection sampling algorithm

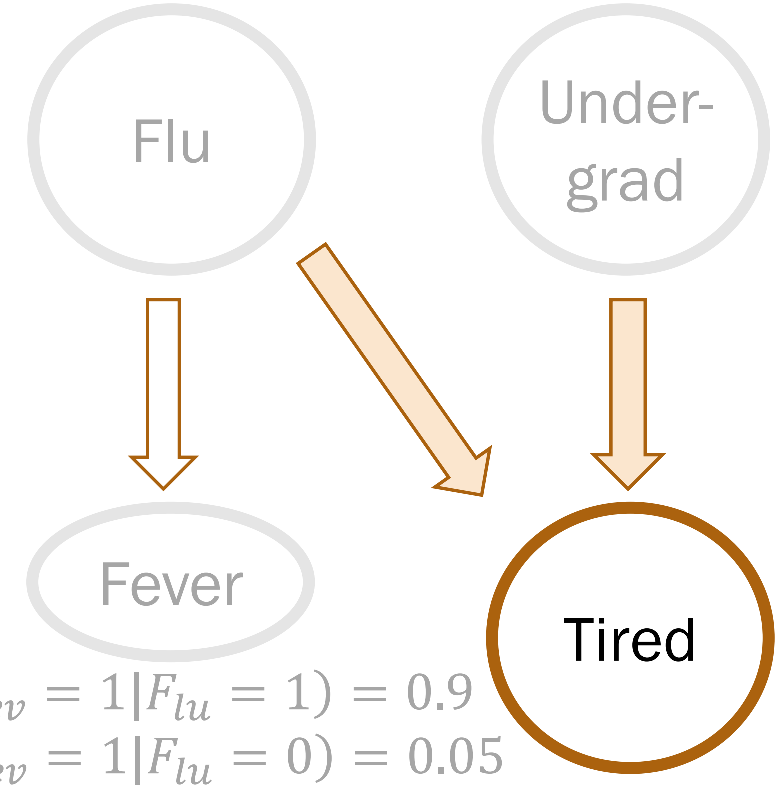
```python
# Method: Make Sample
# --------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

    # choose tired based on (undergrad and flu)
    #
    # TODO: fill in
    #
    #

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

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




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

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

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

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

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

$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

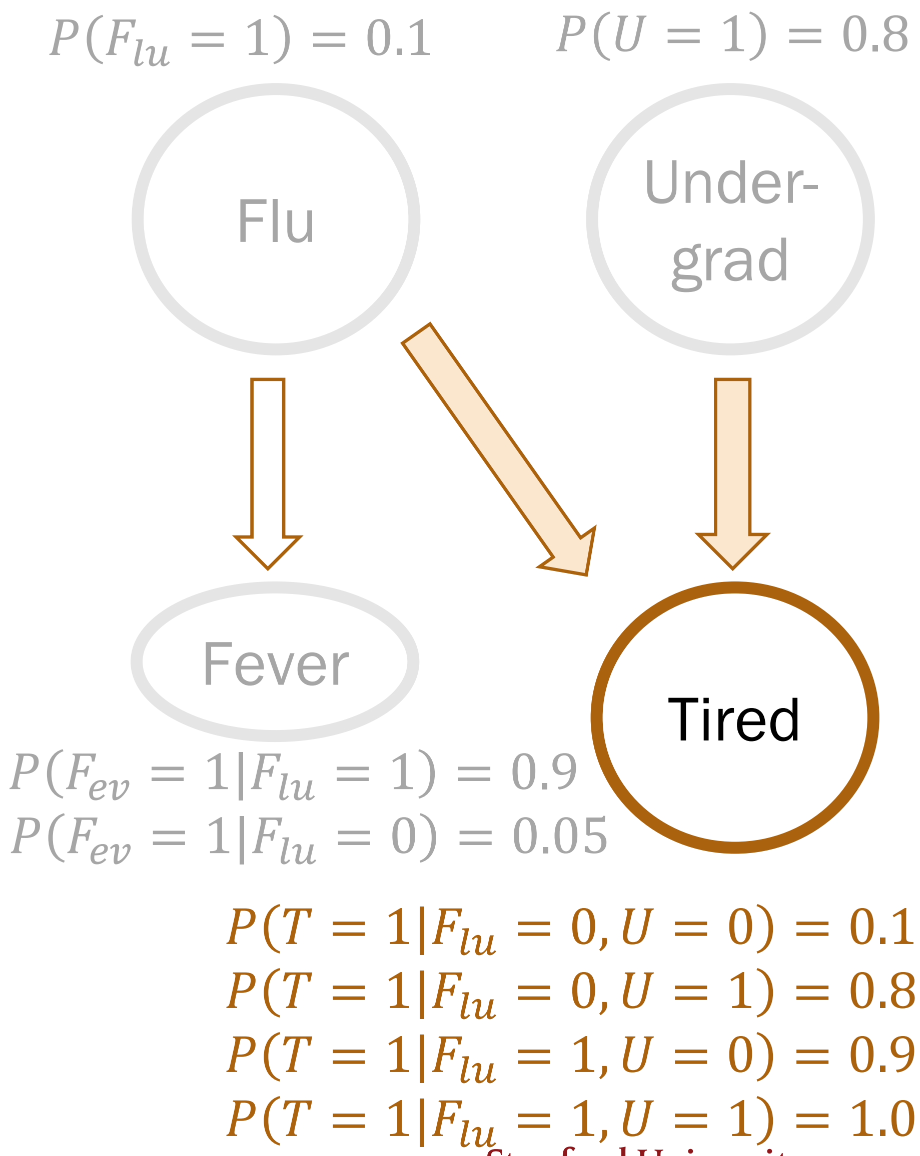
# Rejection sampling algorithm

```python
# Method: Make Sample
# -------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

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

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



# Rejection sampling algorithm

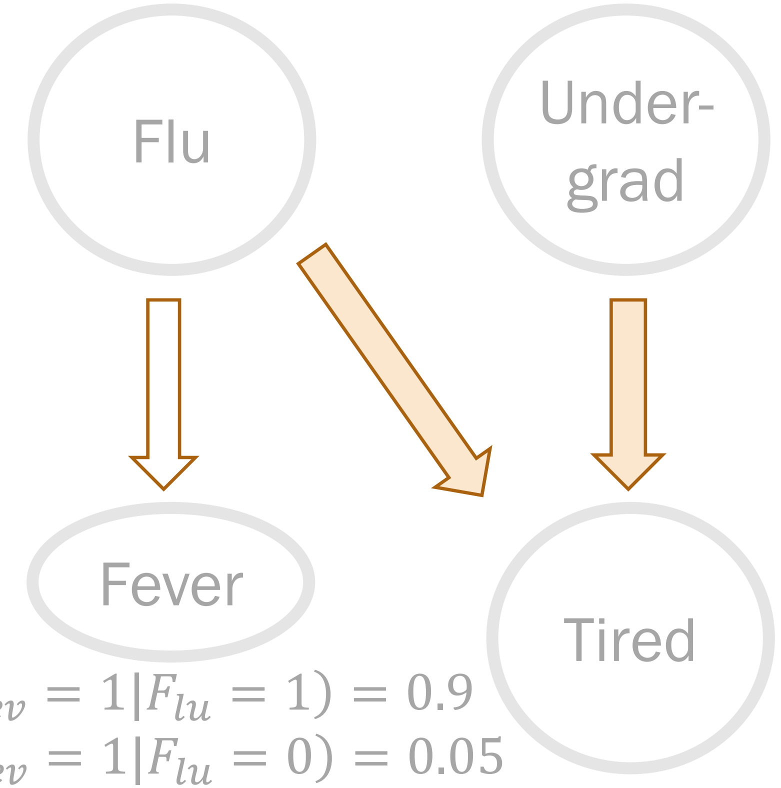
```
# Method: Make Sample
# ---------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

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

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

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

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




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

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

# Interlude for jokes/announcements

# Announcements

**Problem Set 4**

Out: Today!
Due: Monday 10/26 1pm
Covers: Up to and including today

**Mid-quarter feedback form**

link

Open until: next Friday

**Python tutorial #3**

When: Mon 10/19 6-7**pm** PT
Recorded? Yes
Covers: PS4-PS6 content
Notes: to be posted online

# Announcements: CS109 contest

Do something cool and creative with probability

**Grand Prize:**

**Two lowest quizzes replaced with 100%**

**Finalists:**

**Lowest quiz replaced with 100%**

Optional Proposal: Mon. 11/2, 11:59pm
Due: Sat. 11/14, 11:59pm

https://web.stanford.edu/class/cs109/psets/cs109_contest.pdf

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation = …
        # number of samples with (U = 1, T = 1)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T
  return len(samples_event)/len(samples_observ
```

[flu, und, fev, tir]

```
Sampling...
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 1, 1]
[0, 1, 0, 0]
[1, 1, 1, 1]
[0, 0, 1, 1]
...
[0, 1, 0, 1]
Finished sampling
```

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation = …
        # number of samples with (U = 1, T = 1)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T = 1)
  return len(samples_event)/len(samples_observation)
```

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation =
        reject_inconsistent(samples, observation)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T = 1)
  return len(samples_event)/len(samples_observation)
```

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
  samples = sample_a_ton()
  samples_observation =
        reject_inconsistent(samples, observation)
  samples_event =
        # number of samples with (F_lu = 1, U = 1, T = 1)
  return len(samples_event)/len(samples_observation)
```

Keep only samples that are consistent with the observation $(U = 1, T = 1)$.

# Rejection sampling algorithm

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

```python
def rejection_sampling(event, observation):
    samples = sample_a_ton()
    samples_observation =
            reject_inconsistent(samples, observation)
    samples_

    return
```

```python
# Method: Reject Inconsistent
# -------------------
# Rejects all samples that do not align with the outcome.
# Returns a list of consistent samples.
def reject_inconsistent(samples, outcome):
    consistent_samples = []
    for sample in samples:
        if check_consistent(sample, outcome):
            consistent_samples.append(sample)
    return consistent_samples
```

$(U = 1, T = 1)$

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
    samples = sample_a_ton()
    samples_observation =
            reject_inconsistent(samples, observation)
    samples_event =
            reject_inconsistent(samples_observation, event)
    return len(samples_event)/len(samples_observation)
```

Conditional event = samples with $(F_{lu} = 1, U = 1, T = 1)$.

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
    samples = sample_a_ton()
    samples_observation =
            reject_inconsistent(samples, observation)
    samples_event =
            reject_inconsistent(samples_observation, event)
    return l
```

```
def reject_inconsistent(samples, outcome):
```

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

```
    return consistent_samples
```

ation)

Conditi

$= 1).$

# Rejection sampling algorithm

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

```
def rejection_sampling(event, observation):
    samples = sample_a_ton()
    samples_observation =
            reject_inconsistent(samples, observation)
    samples_event =
            reject_inconsistent(samples_observation, event)
    return len(samples_event)/len(samples_observation)
```

Approximate Probability = $\dfrac{\text{\# samples with } (F_{lu} = 1, U = 1, T = 1)}{\text{\# samples with } (U = 1, T = 1)}$

# To the code!

# Rejection sampling

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

With enough samples, you can correctly compute:

- Probability estimates
- Conditional probability estimates
- Expectation estimates

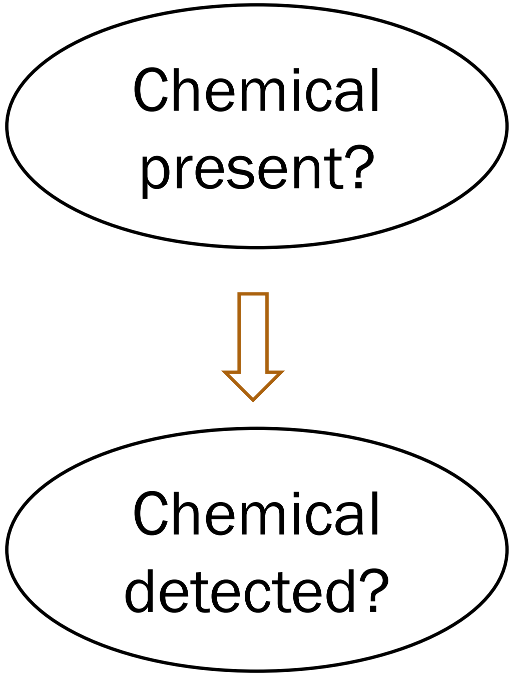
Because your samples are a representation of the joint distribution!

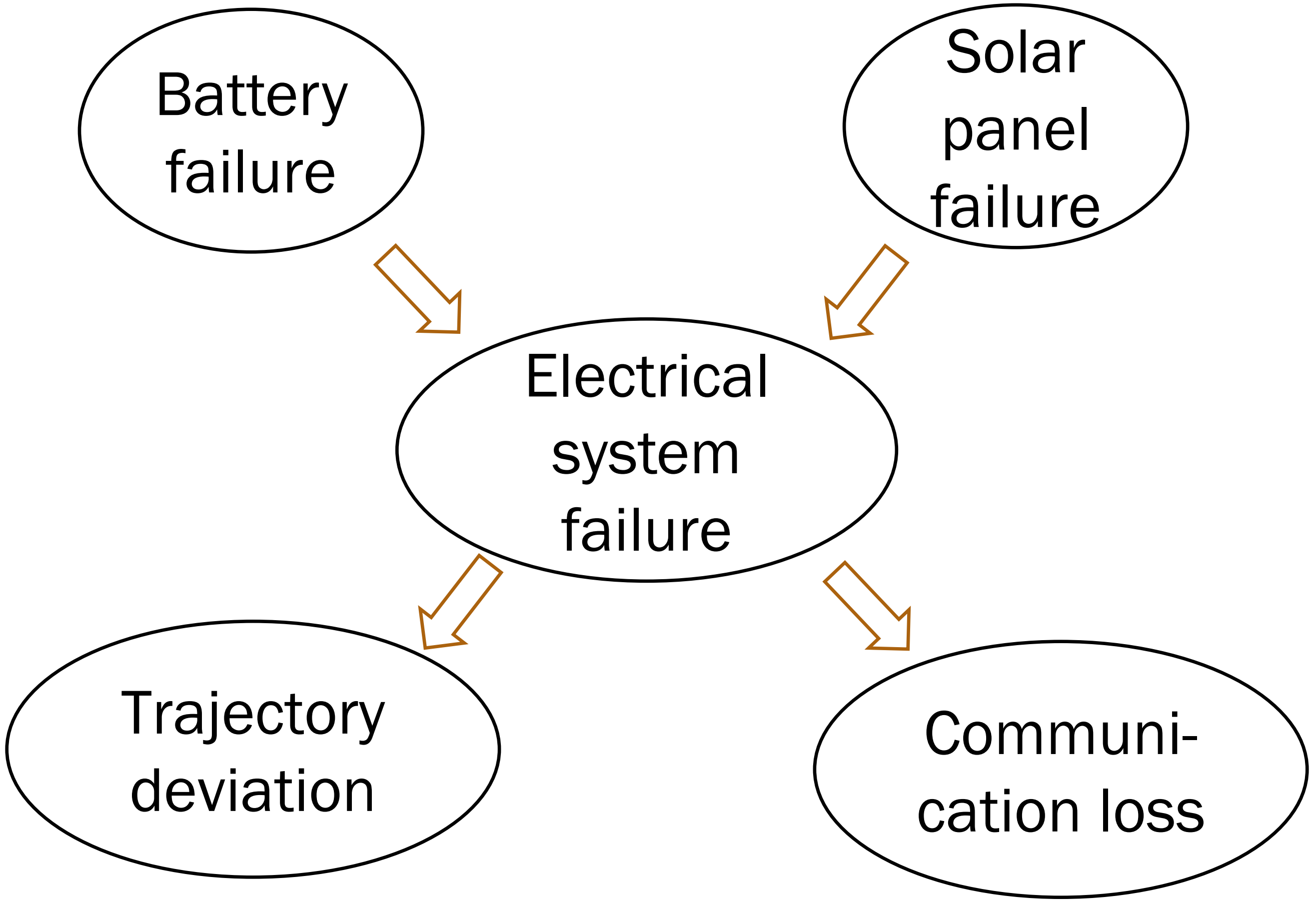
`[flu, und, fev, tir]`

```
Sampling...
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 1, 0, 1]
[0, 0, 0, 0]
[0, 1, 0, 1]
[0, 1, 1, 1]
[0, 1, 0, 0]
[1, 1, 1, 1]
[0, 0, 1, 1]
...
[0, 1, 0, 1]
Finished sampling
```

`P(has flu | undergrad and is tired) = 0.122`

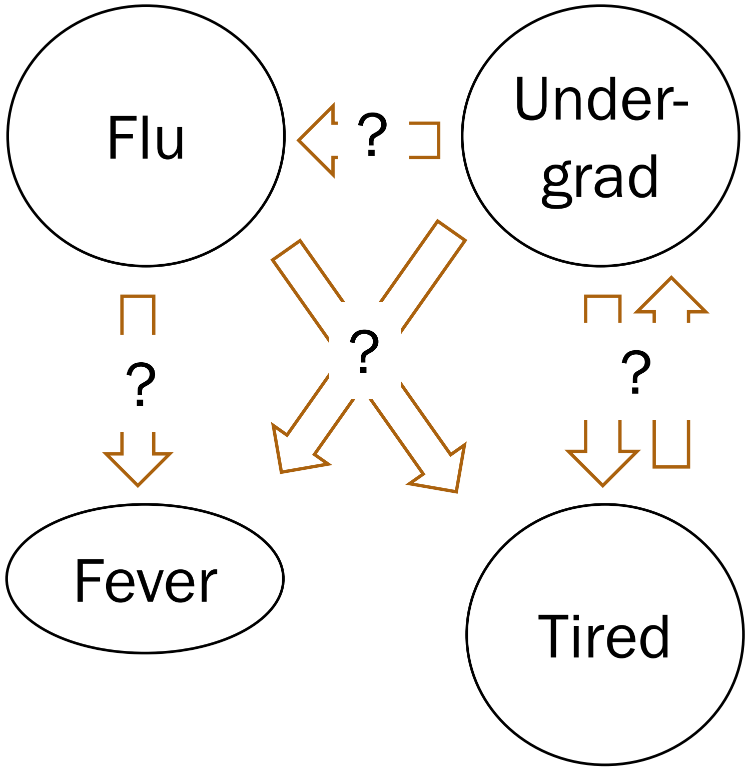
# Other applications




Take CS238/AA228: Decision Making under Uncertainty!

# Challenge with Bayesian Networks




What if we don't know the structure?

Take CS228: Probabilistic Graphical Models!

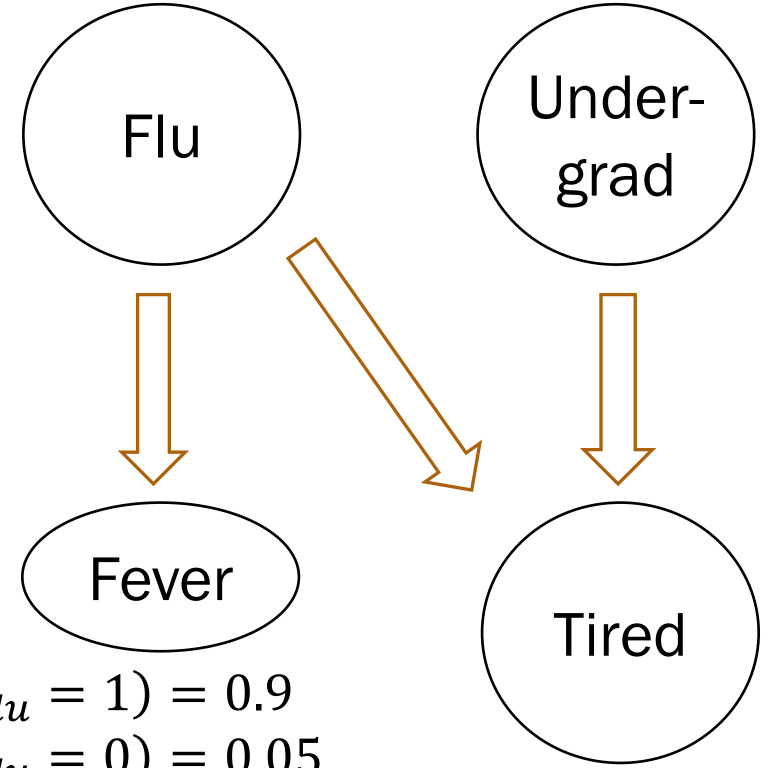
# Disadvantages of rejection sampling

$$P(F_{lu} = 1 | F_{ev} = 1)?$$

What if we never encounter some samples?

```
[flu=0, und, fev=1, tir]
```

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




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

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

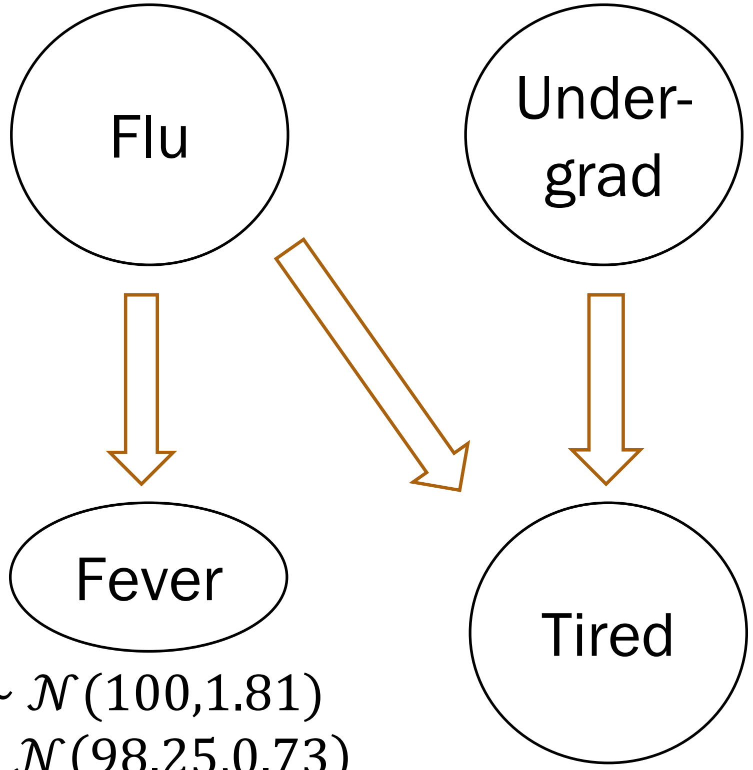
# Disadvantages of rejection sampling

$$P(F_{lu} = 1 | F_{ev} = 99.4)?$$

What if we never encounter some samples?

What if random variables are continuous?

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




$F_{ev} | F_{lu} = 1 \sim \mathcal{N}(100, 1.81)$

$F_{ev} | F_{lu} = 0 \sim \mathcal{N}(98.25, 0.73)$

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

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

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

$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

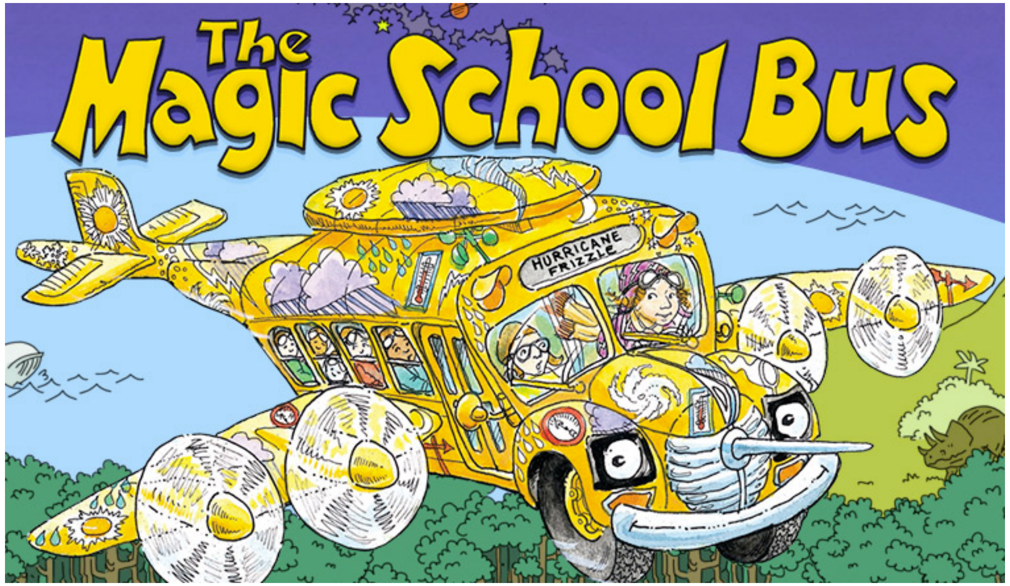
(no video)

# Gibbs sampling (extra)

# Gibbs Sampling (not covered)

Basic idea:

- Fix all observed events
- Incrementally sample a new value for each random variable
- Difficulty: More coding for computing different posterior probabilities




Learn in extra slides/extra notebook!

(or by taking CS228/CS238)