20: Inference

David Varodayan

February 24, 2020

Adapted from slides by Lisa Yan

Today's plan

Bootstrapping

- For a statistic
- For a p-value

Definition: Bayesian Networks

Inference:

1. Math
2. Rejection sampling ("joint" sampling)

Null hypothesis test

Nepal Happiness
4.44
3.36
5.87
2.31
…
3.70

Bhutan Happiness
4.44
3.36
5.87
2.31
…
3.70

$\mu_1 = 3.1$ $\mu_2 = 2.4$

Claim: The difference in mean happiness between Nepal and Bhutan is 0.7 happiness points.

Null hypothesis test

def **null hypothesis** – Even if there is no pattern (i.e., the two samples are from identical distributions), your claim might have arisen by chance.

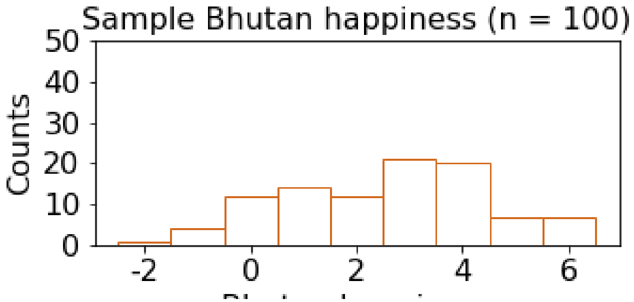
def **p-value** – What is the probability that, under the null hypothesis, the observed difference occurs?

Claim: The difference in mean happiness between Nepal and Bhutan is 0.7 happiness points.

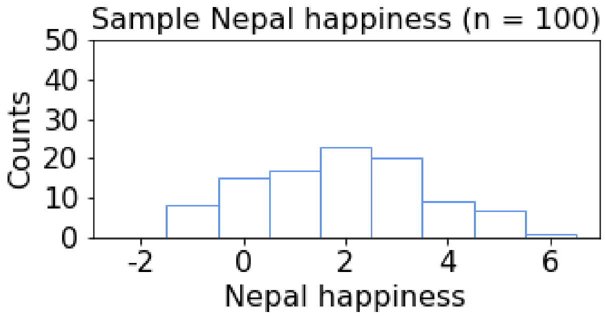
Bootstrap for p-values

1. Create a universal sample using your two samples

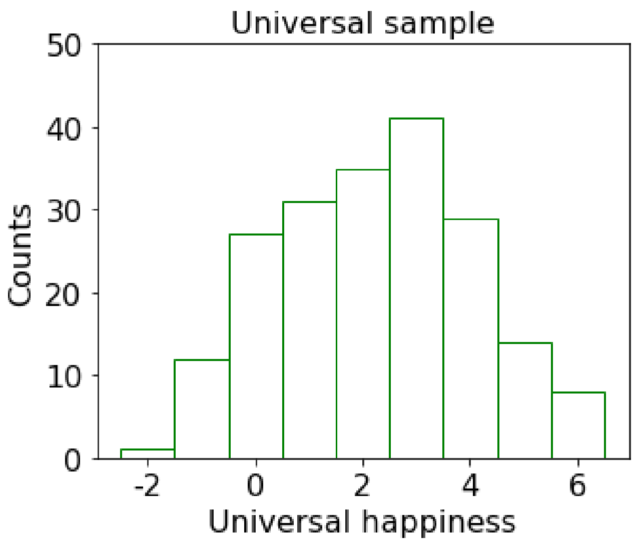
Recreate the null hypothesis



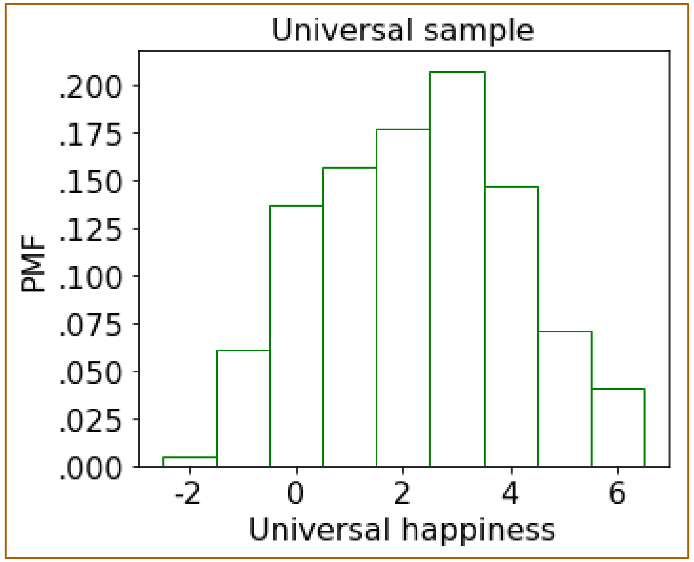
+



=



≈



Bootstrap for p-values

1. Create a universal sample using your two samples
2. Repeat **10,000** times:
    a. Resample **both samples**
    b. Recalculate the **mean absolute difference** between the resamples

3. **p-value** = $\dfrac{\text{\# (mean absolute diff >= observed absolute diff)}}{10{,}000}$

Probability that observed difference arose by chance

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   observed_diff = |mean of bhutan_sample – mean of nepal_sample|

   uni_sample = combine bhutan_sample and nepal_sample
   count = 0

   repeat 10,000 times:
      bhutan_resample = draw N resamples from the uni_sample
      nepal_resample = draw M resamples from the uni_sample
      muBhutan = sample mean of the bhutan_resample
      muNepal = sample mean of the nepal_resample
      diff = |muNepal - muBhutan|
      if diff >= observed_diff:
         count += 1

pValue = count / 10,000
```

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   observed_diff = |mean of bhutan_sample – mean of nepal_sample|

   uni_sample = combine bhutan_sample and nepal_sample
   count = 0

   repeat 10,000 times:
      bhutan_resample = draw N resamples from the uni_sample
      nepal_resample = draw M resamples from the uni_sample
      muBhutan = sample mean of the bhutan_resample
      muNepal = sample mean of the nepal_resample
      diff = |muNepal - muBhutan|
      if diff >= observed_diff:
         count += 1

pValue = count / 10,000
```

1. Create a universal sample using your two samples

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   observed_diff = |mean of bhutan_sample – mean of nepal_sample|

   uni_sample = combine bhutan_sample and nepal_sample
   count = 0

   repeat 10,000 times:
      bhutan_resample = draw N resamples from the uni_sample
      nepal_resample = draw M resamples from the uni_sample
      muBhutan = sample mean of the bhutan_resample
      muNepal = sample mean of the nepal_resample
      diff = |muNepal - muBhutan|
      if diff >= observed_diff:
         count += 1

pValue = count / 10,000
```

2. a. Resample **both samples with replacement**

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
   N = size of the bhutan_sample
   M = size of the nepal_sample
   observed_diff = |mean of bhutan_sample – mean of nepal_sample|

   uni_sample = combine bhutan_sample and nepal_sample
   count = 0

   repeat 10,000 times:
      bhutan_resample = draw N resamples from the uni_sample
      nepal_resample = draw M resamples from the uni_sample
      muBhutan = sample mean of the bhutan_resample
      muNepal = sample mean of the nepal_resample
      diff = |muNepal - muBhutan|
      if diff >= observed_diff:
         count += 1

pValue = count / 10,000
```

2. b. Recalculate the **mean difference** between resamples

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample – mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

pValue = count / 10,000
```

3. **p-value** =

$$\frac{\text{\# (mean diffs >= observed diff)}}{10{,}000}$$

Bootstrap

Try it yourself!

http://web.stanford.edu/class/cs109/demos/bootstrap.zip

Null hypothesis test

Nepal Happiness
4.44
3.36
5.87
2.31
…
3.70

$\mu_1 = 3.1$

Bhutan Happiness
4.44
3.36
5.87
2.31
…
3.70

$\mu_2 = 2.4$

Claim: The happiness of Nepal and Bhutan are from different distributions with a 0.7 difference of means ($p < 0.01$).

Announcements

Problem Set 5

Due: Friday 2/28
Covers: Up to Lecture 19

Late Day Reminder

No late days permitted past last day of the quarter, 3/13

CS109 Contest

Due: Monday 3/9 11:59pm

A small change in gears

Bootstrapping – Use code to compute statistics when you only have data, not the underlying distribution.

What if you have the underlying distribution of joint random variables (via an expert), but finding closed forms of joint probabilities is intractable?

Today's plan

Bootstrapping

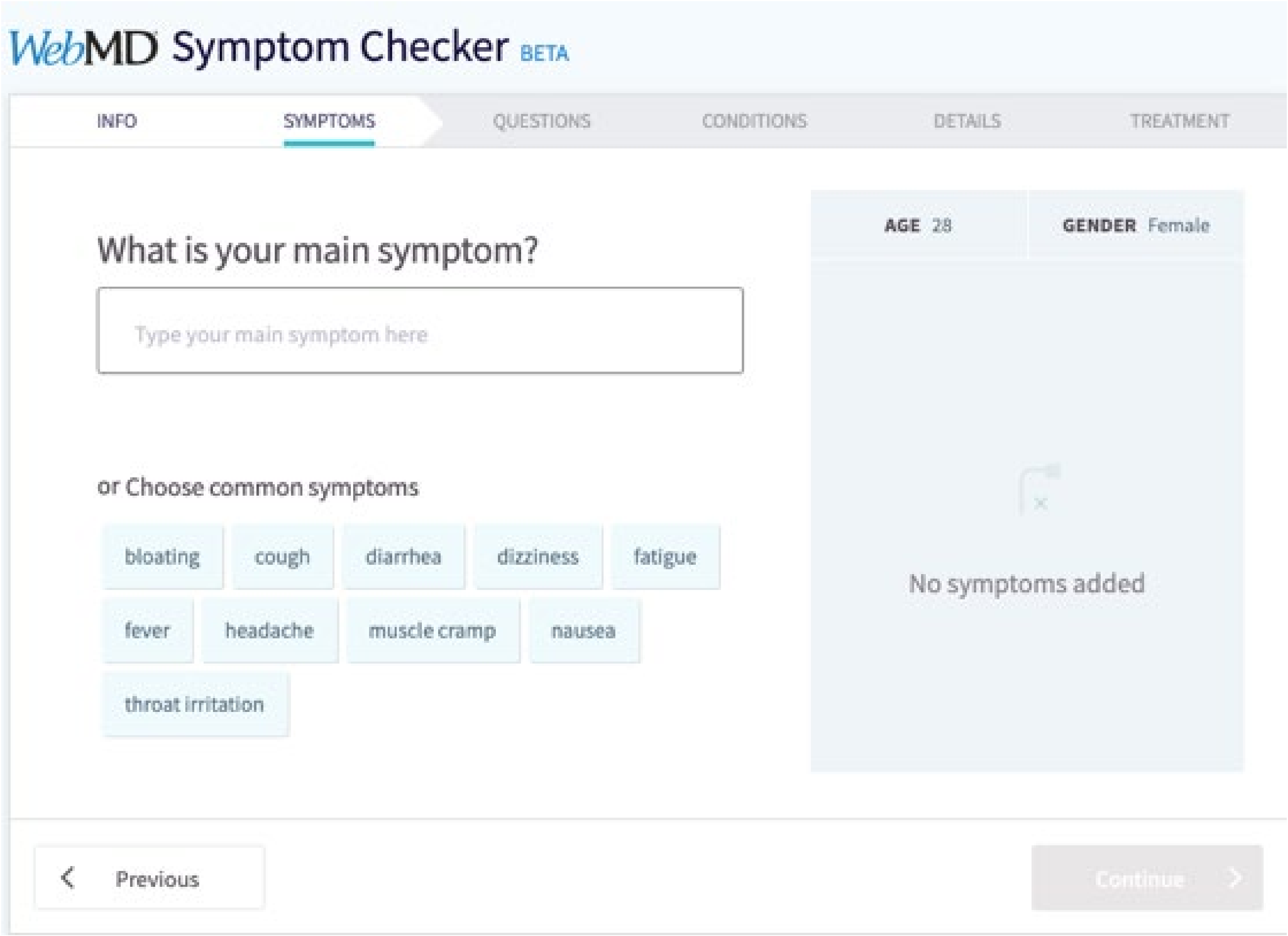
- For a statistic
- For a p-value

Definition: Bayesian Networks

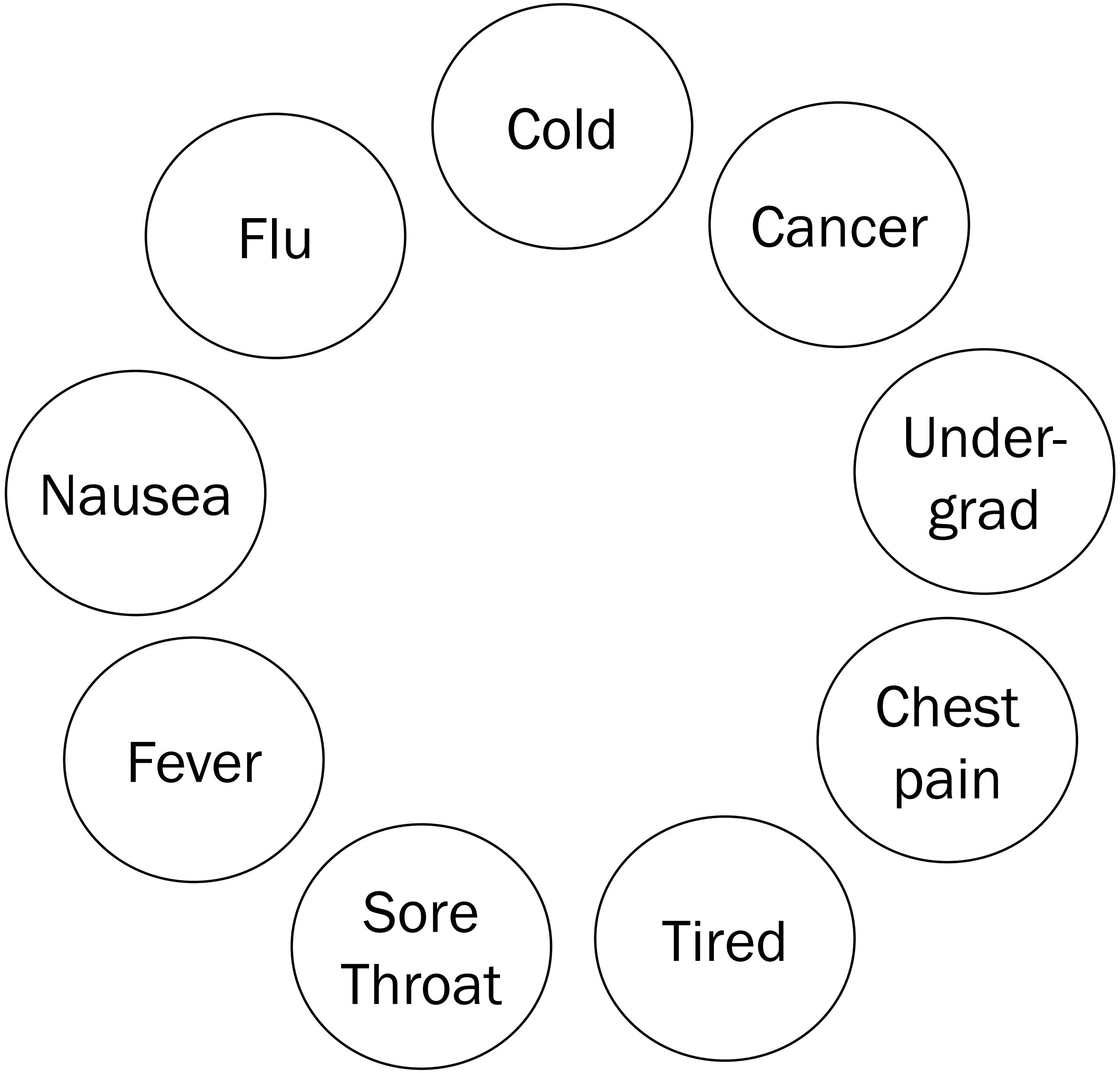
Inference:

1. Math
2. Rejection sampling ("joint" sampling)
3. Optional: Gibbs sampling (MCMC algorithm)

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

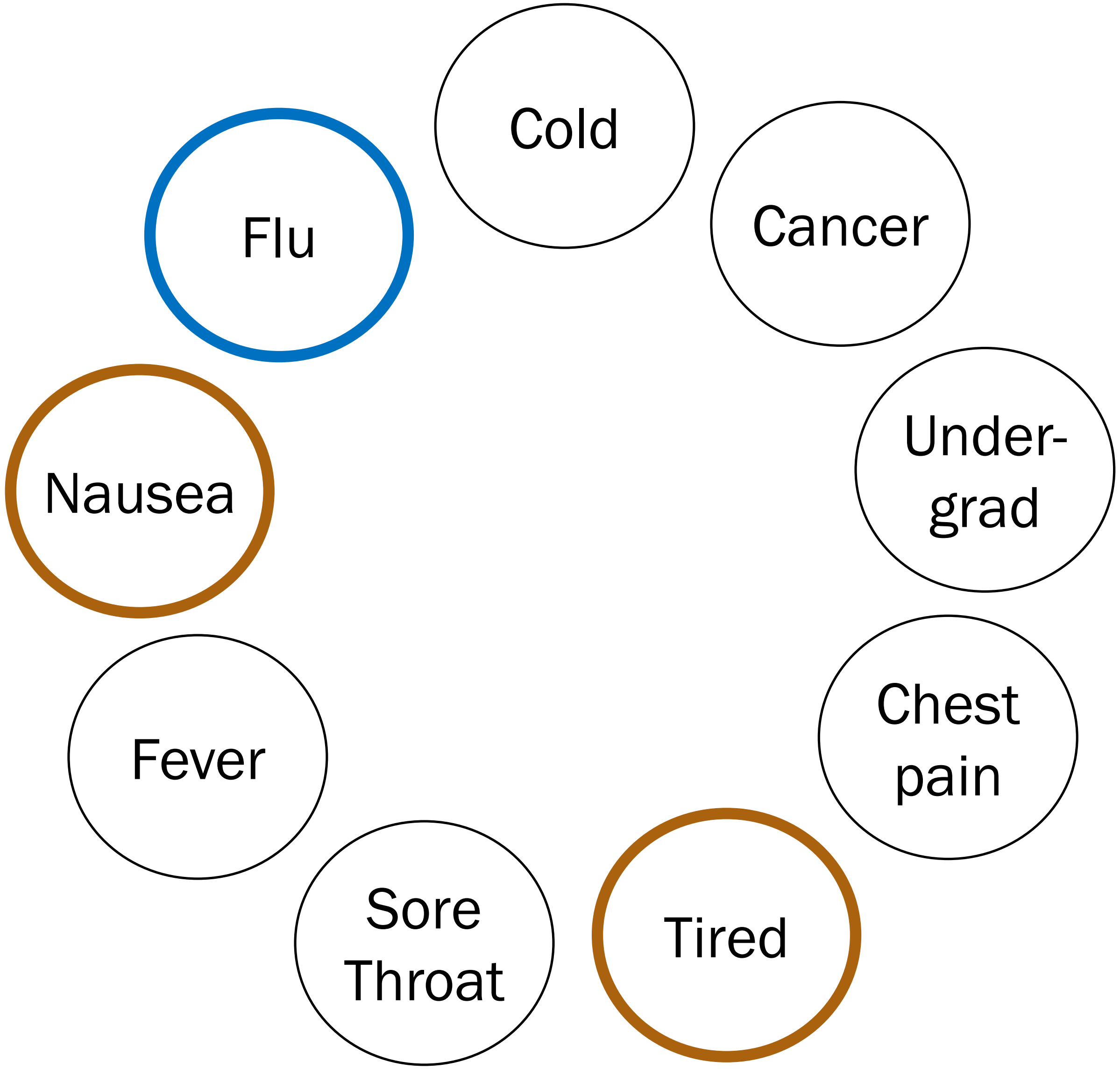
Inference



General inference question:

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

Inference

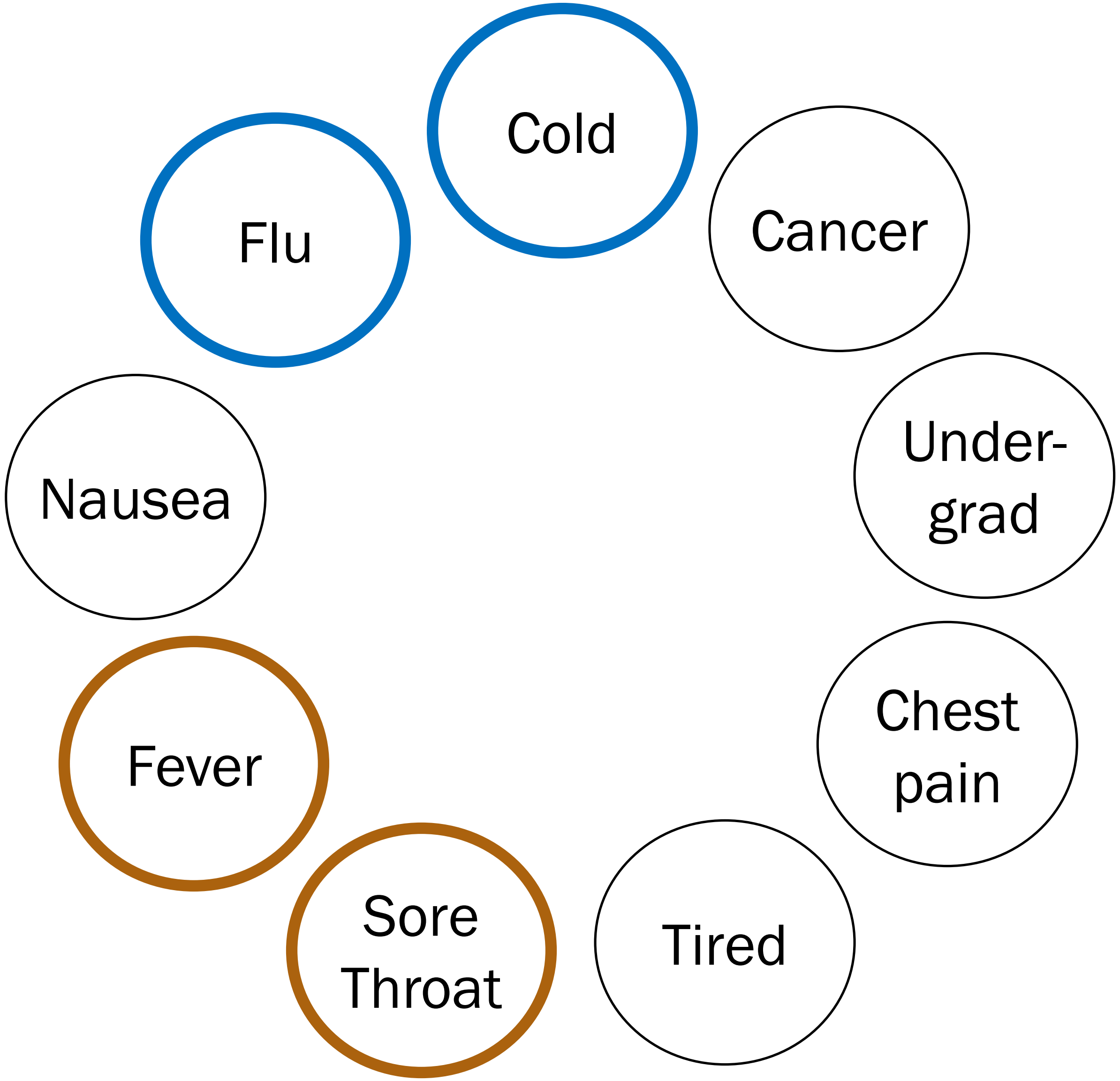


One inference question:

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

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

Inference

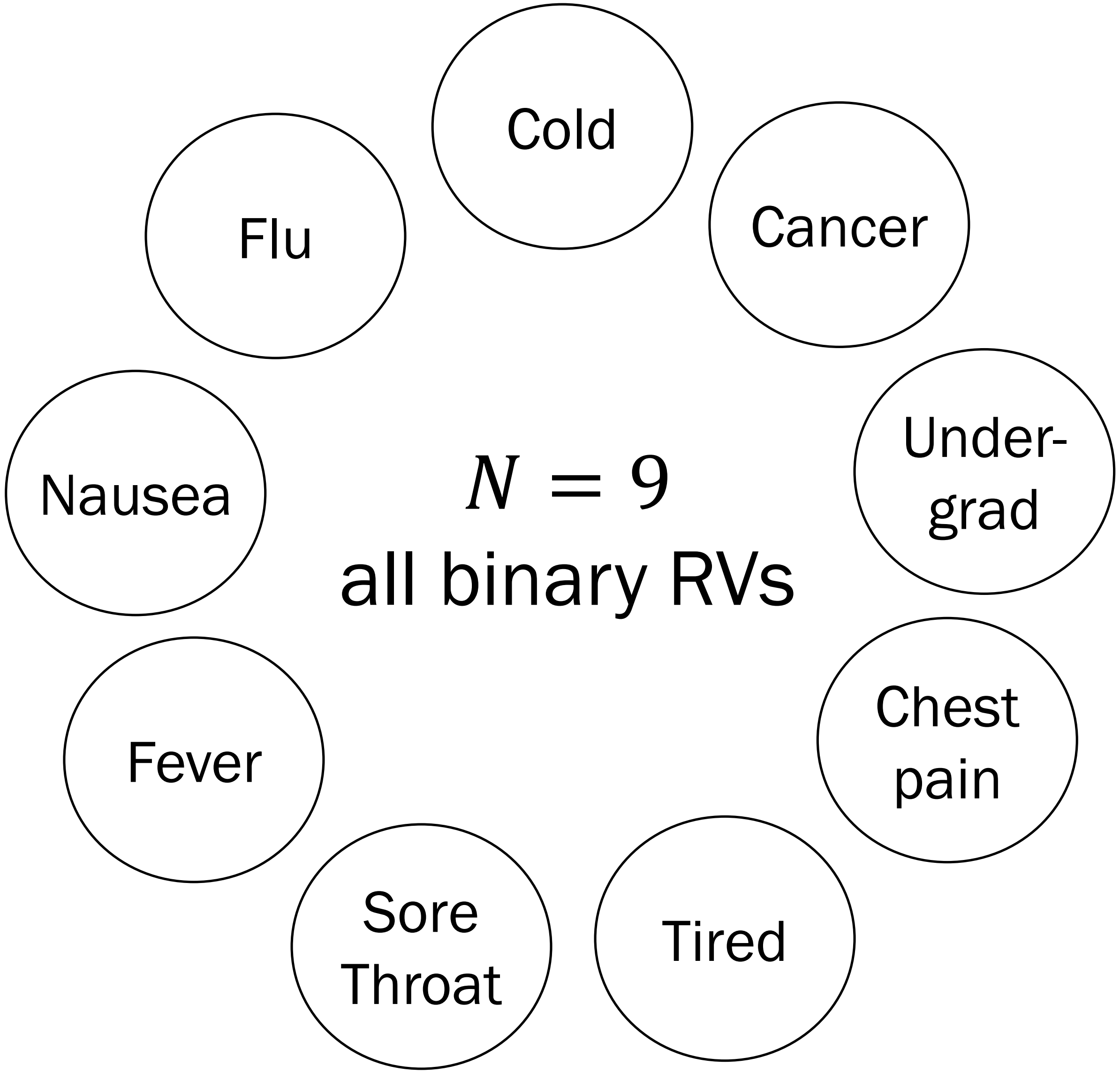


<u>Another inference question</u>:

$$P(C_o = 1, F_{lu} = 1 | S = 0, F_e = 0)$$

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

Inference

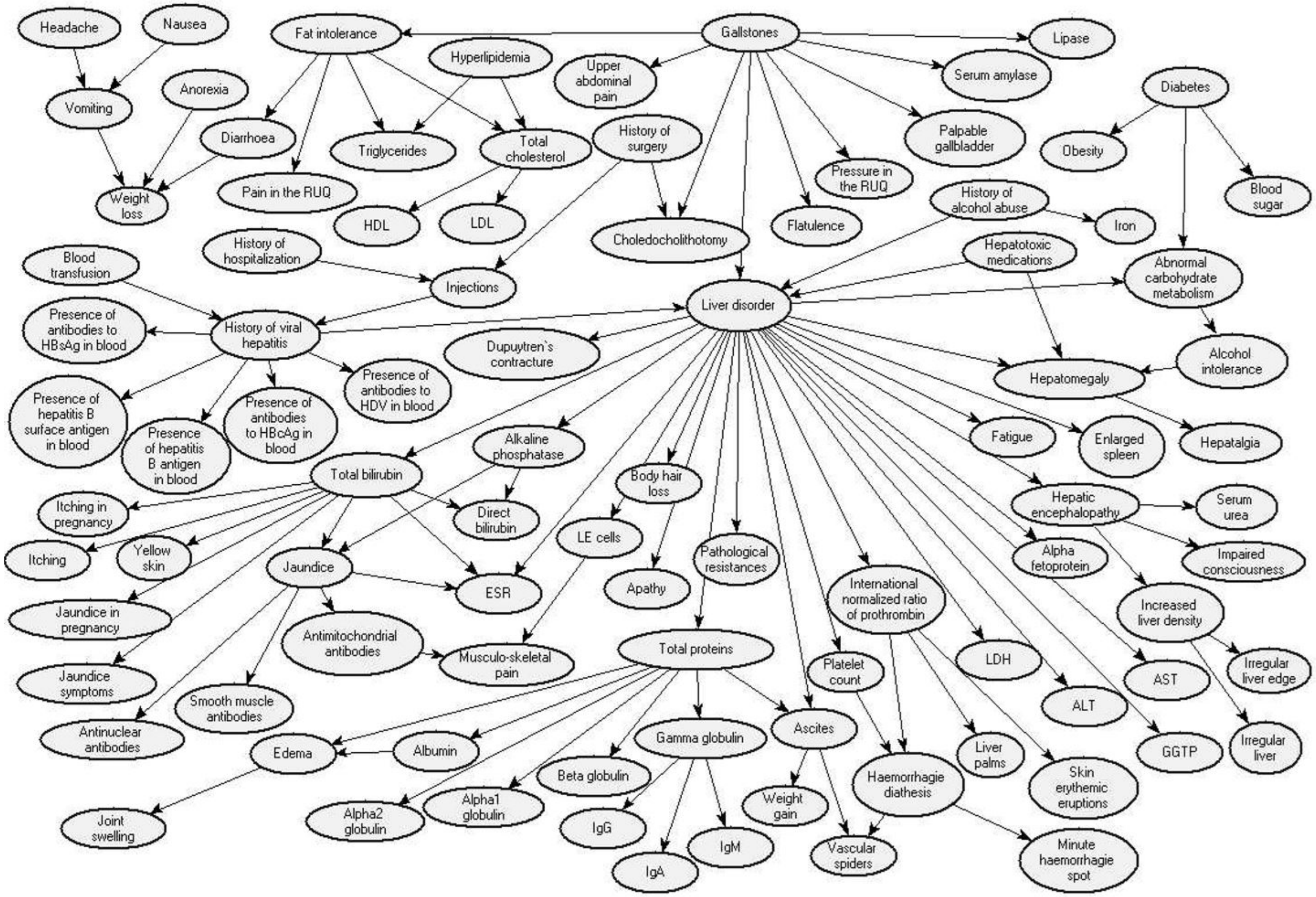


If we knew the **joint distribution**, we can answer all probabilistic inference questions.

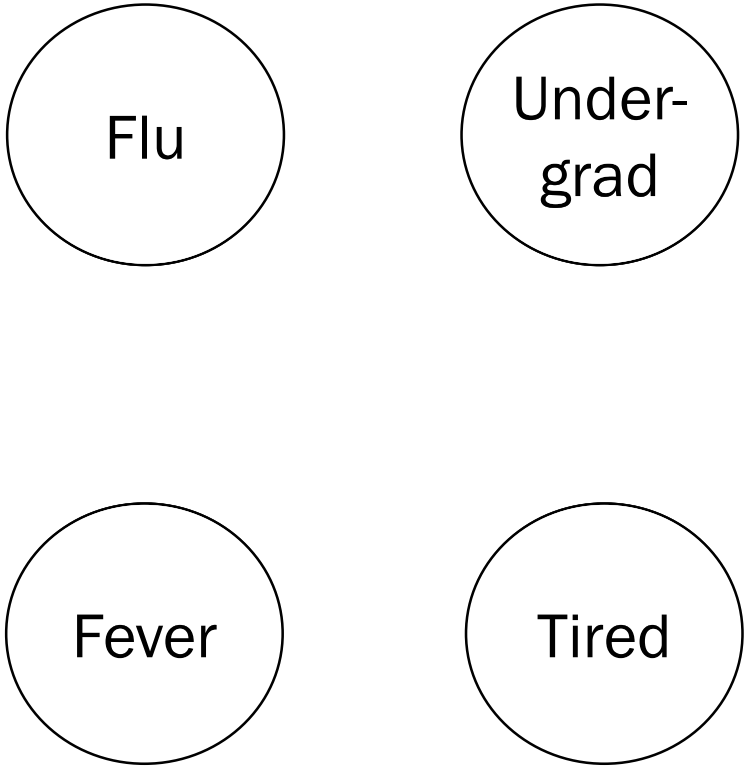
What is the size of the joint probability table?

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

N can be large…



A simpler WebMD



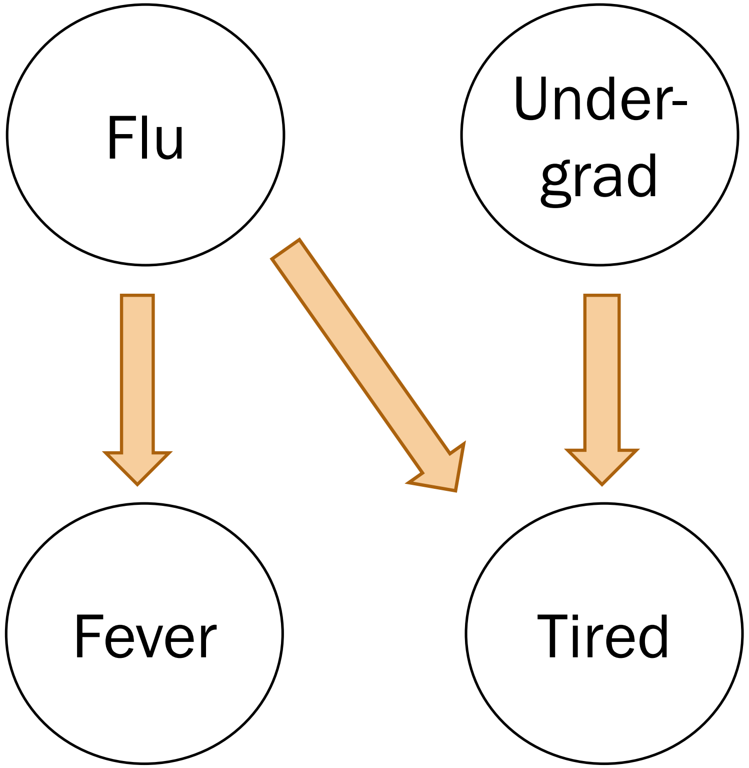
Great! Just specify $2^4 = 16$ joint probabilities…?

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

What would a Stanford flu expert do?

A Bayesian Network

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



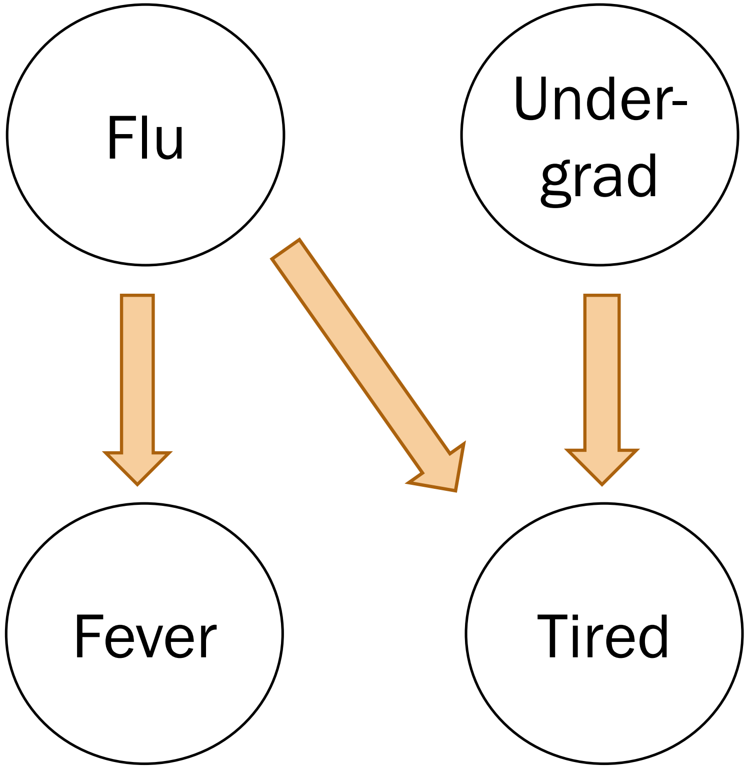
What would a Stanford flu expert do?

1. Describe the joint distribution using causality

A Bayesian Network

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

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



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

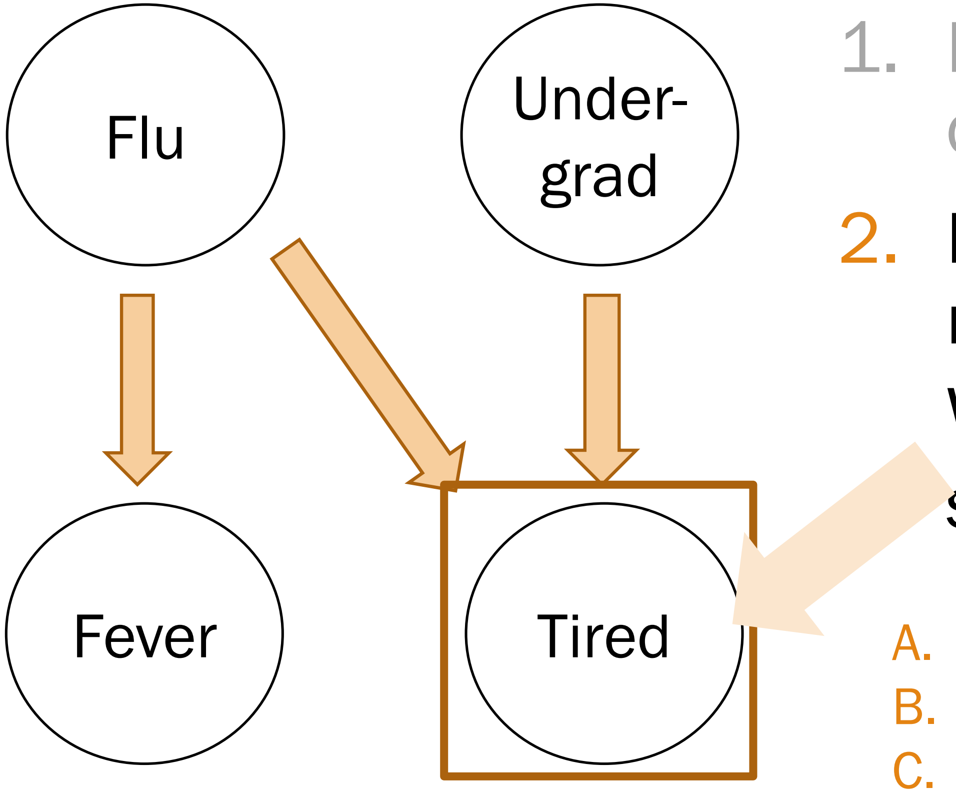
What would a Stanford flu expert do?

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

A Bayesian Network

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

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



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

What would a Stanford flu expert do?

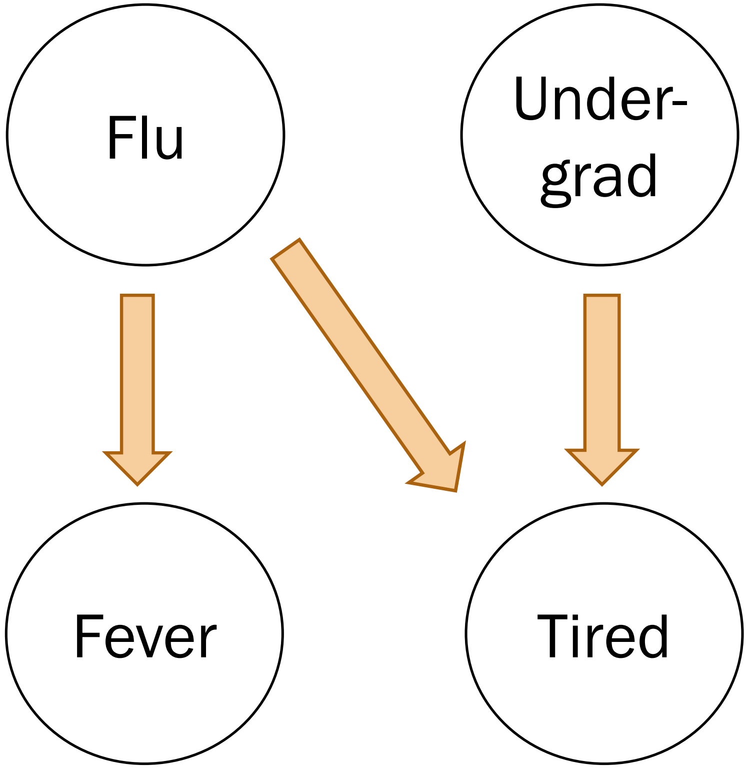
1. Describe the joint distribution using causality
2. Provide P(values|parents) for each random variable

What conditional probabilities should our expert specify? (select all that apply)

A. $P(T = 1|F_{lu} = 0, U = 0)$
B. $P(T = 1|F_{lu} = 0, U = 1)$
C. $P(T = 1|F_{lu} = 1, U = 0)$
D. $P(T = 1|F_{lu} = 1, U = 1)$
E. $P(T = 1|F_{lu} = 0)$
F. $P(T = 1|F_{lu} = 1)$
G. $P(T = 0|F_{lu} = 0, U = 0)$
H. $P(T = 0|F_{lu} = 0, U = 1)$
I. $P(T = 0|F_{lu} = 1, U = 0)$
J. $P(T = 0|F_{lu} = 1, U = 1)$
K. $P(T = 1|U = 0)$
L. $P(T = 1|U = 1)$

A Bayesian Network

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



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

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

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

Today's plan

Bootstrapping

- For a statistic
- For a p-value

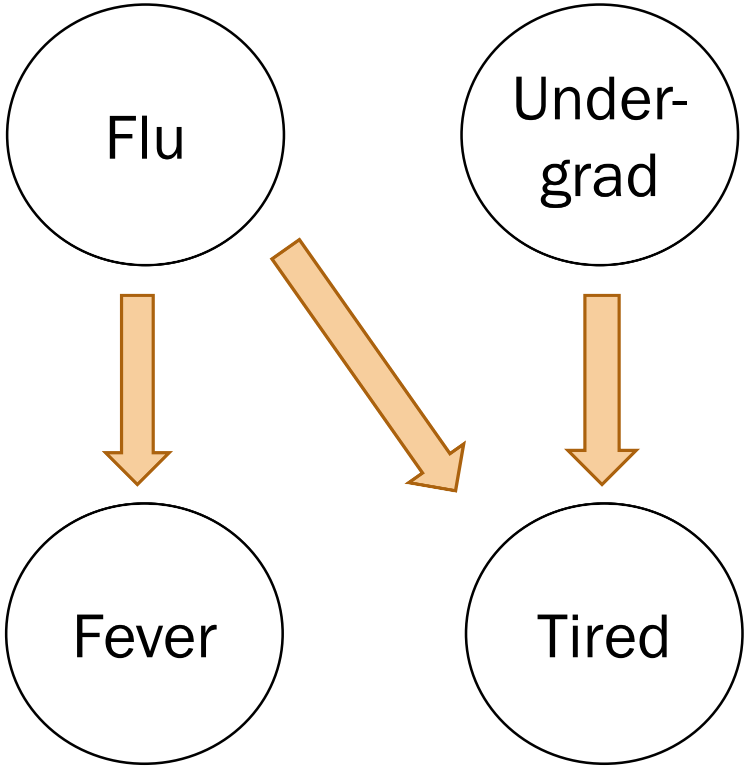
Definition: Bayesian Networks

Inference:

1. Math
2. Rejection sampling ("joint" sampling)

Inference via math

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

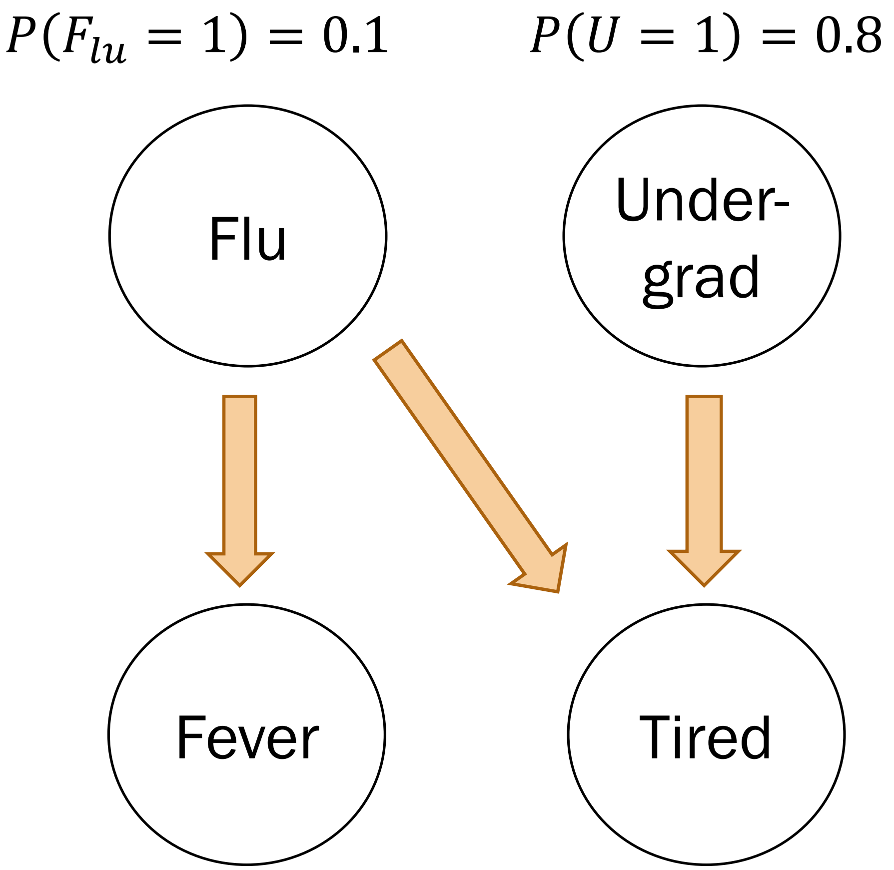


$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

In a Bayesian Network, each random variable is conditionally independent of its non-descendants, given its parents.

Inference via math



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

   Compute joint probabilities using chain rule.

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

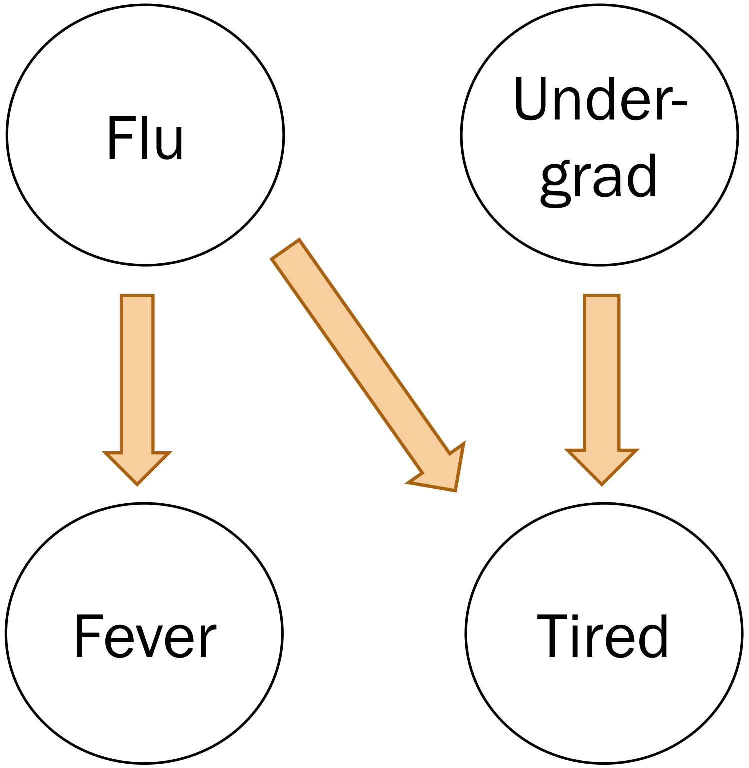
   $$= 0.5472$$

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

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

Inference via math

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



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

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

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

   1. Compute joint probabilities

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

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

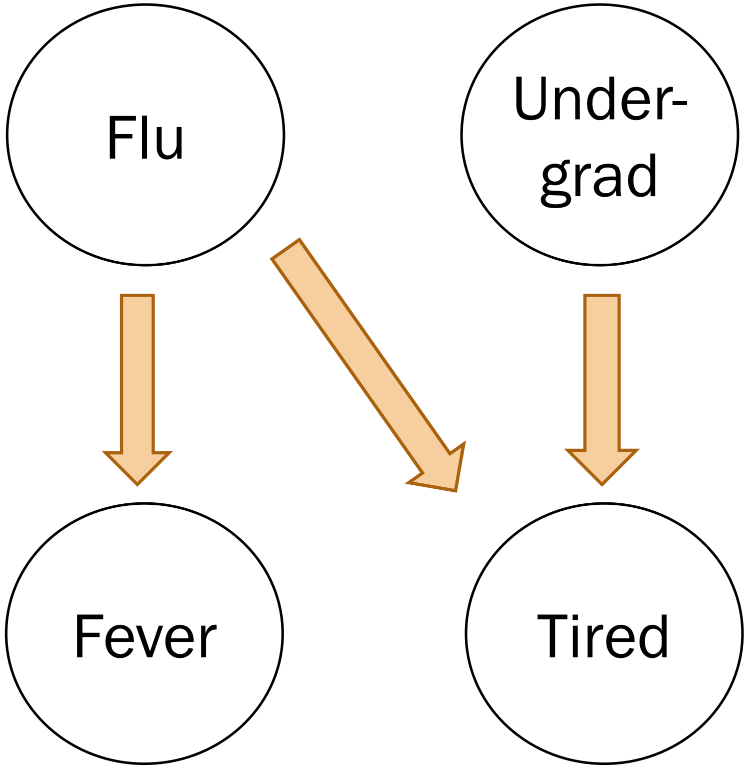
   2. Definition of conditional probability

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

$$= 0.095$$

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

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

      ...

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

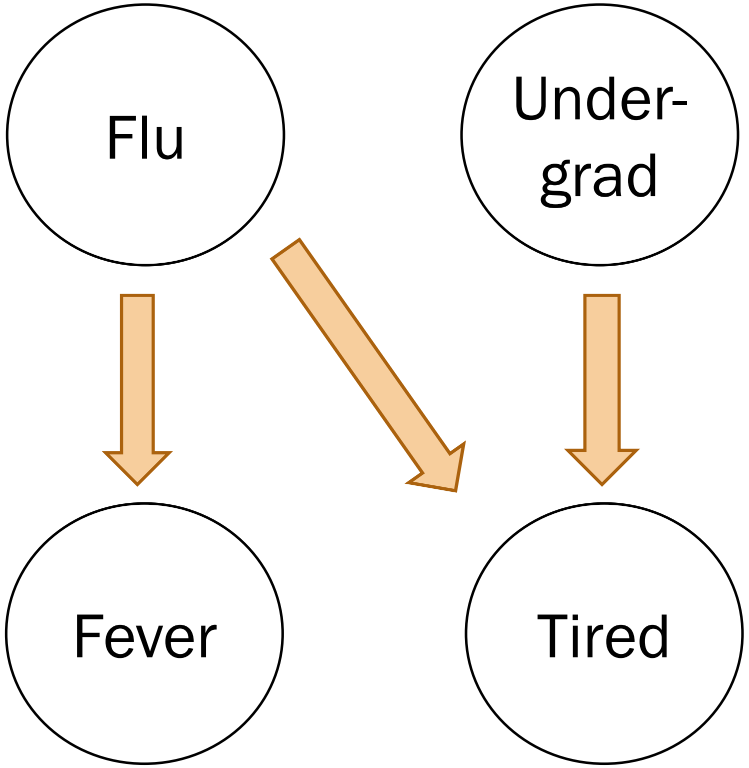
   2. Definition of conditional probability

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

      $$= 0.122$$

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?

Today’s plan

Bootstrapping

- For a statistic
- For a p-value

Definition: Bayesian Networks

Inference:

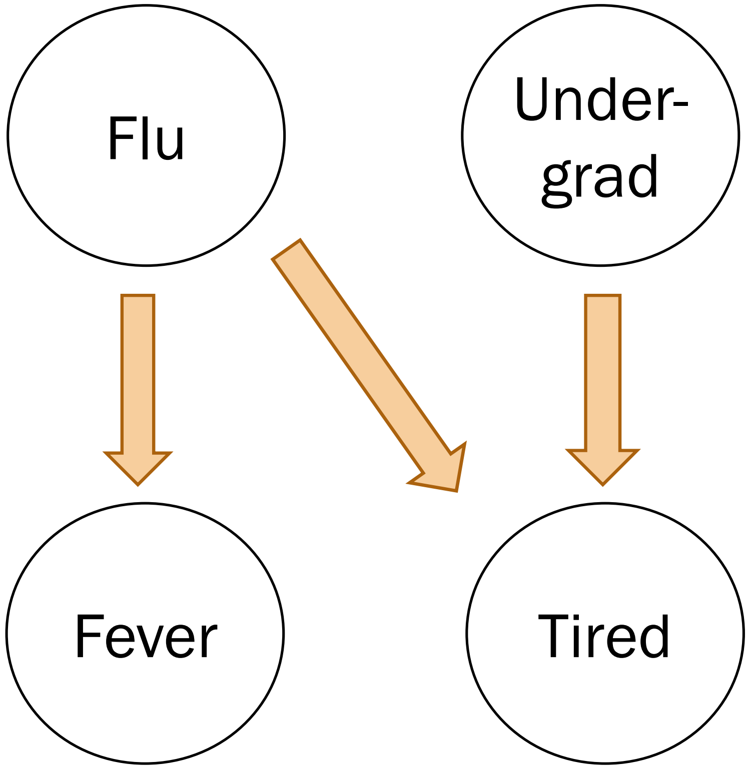
1. Math
2. Rejection sampling (“joint” sampling)

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 =
        reject_inconsistent(samples, observation)
    samples_event =
        reject_inconsistent(samples_observation, event)
    return len(samples_event)/len(samples_observation)
```

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

Rejection sampling algorithm

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

```
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)
```

Rejection sampling algorithm

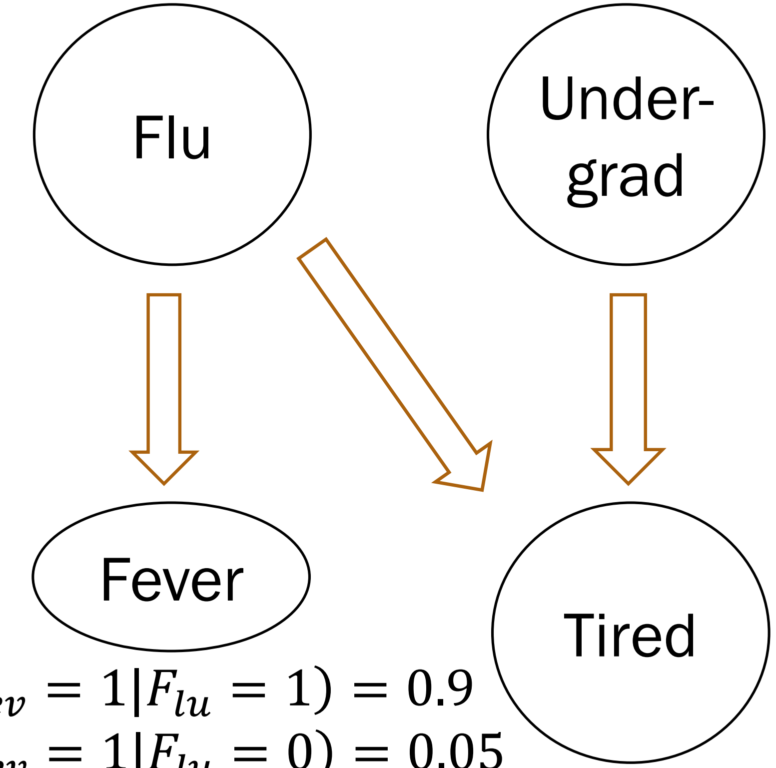
```
N_SAMPLES = 100000
# Method: Sample a ton
# -------------------
# create N_SAMPLES with likelihood proportional
# to the joint distribution
def sample_a_ton():
    samples = []
    for i in range(N_SAMPLES):
        sample = make_sample() # a particle
        samples.append(sample)
    return samples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

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

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) # undergraduate

   # 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]
```

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



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

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

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

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

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

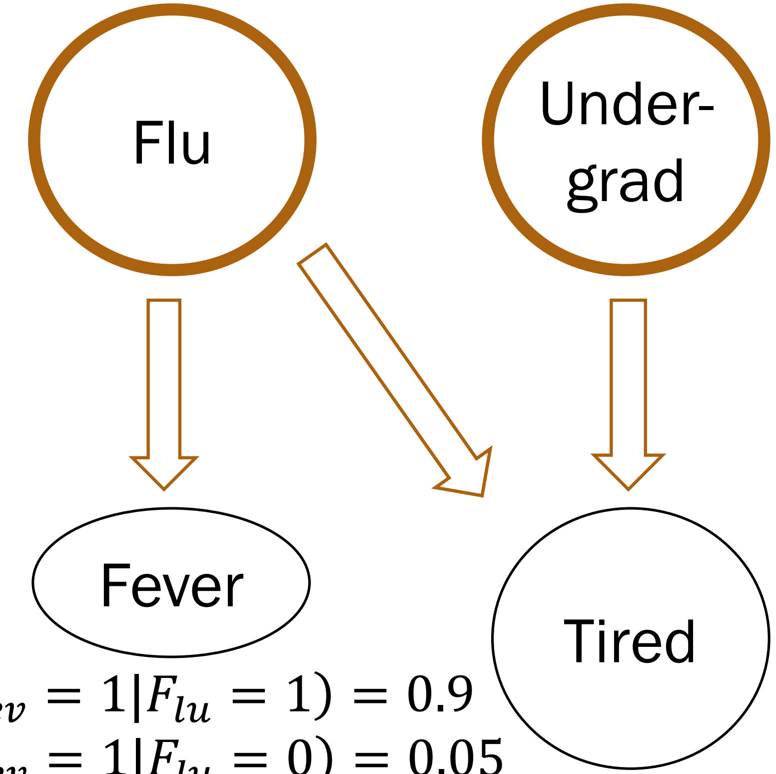
$P(T = 1 | F_{lu} = 1, U = 1) = 1.0$

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) # undergraduate

    # 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]
```

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

$P(U = 1) = 0.8$



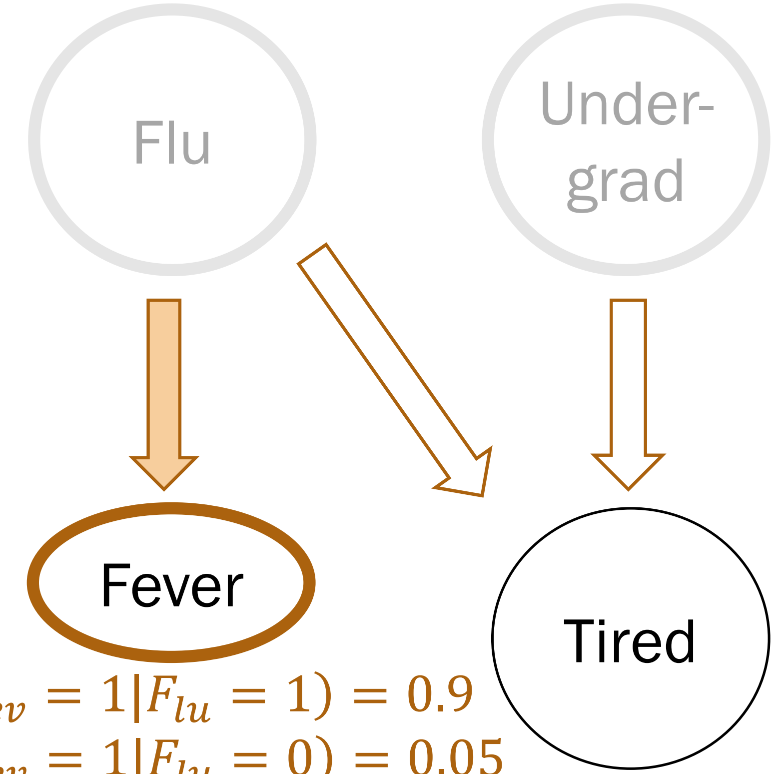
$$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$$
$$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$$

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

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) # undergraduate

    # 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]
```

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



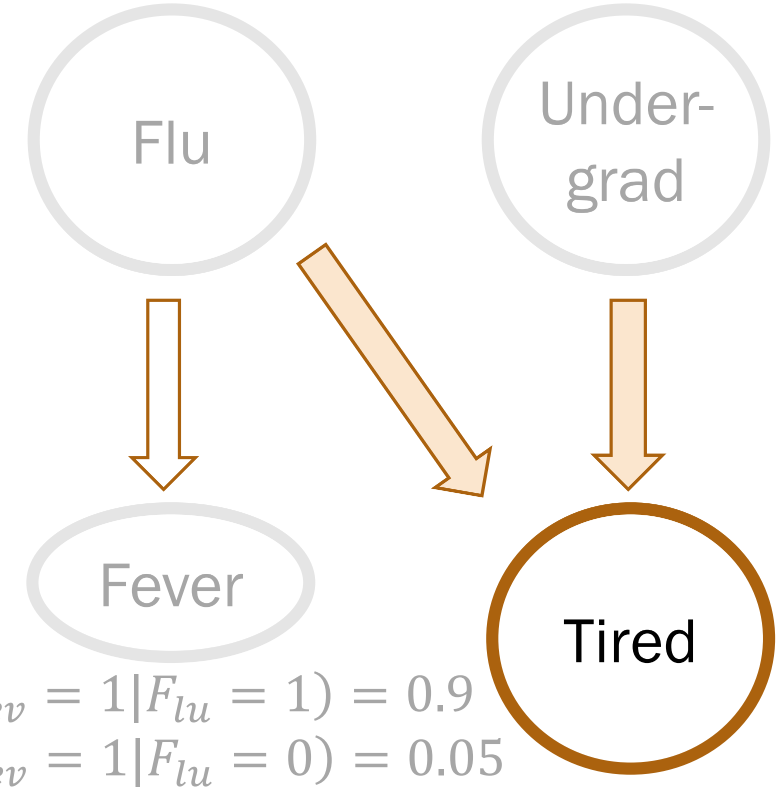
$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

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

Rejection sampling algorithm

```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) # undergraduate

    # 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]
```

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



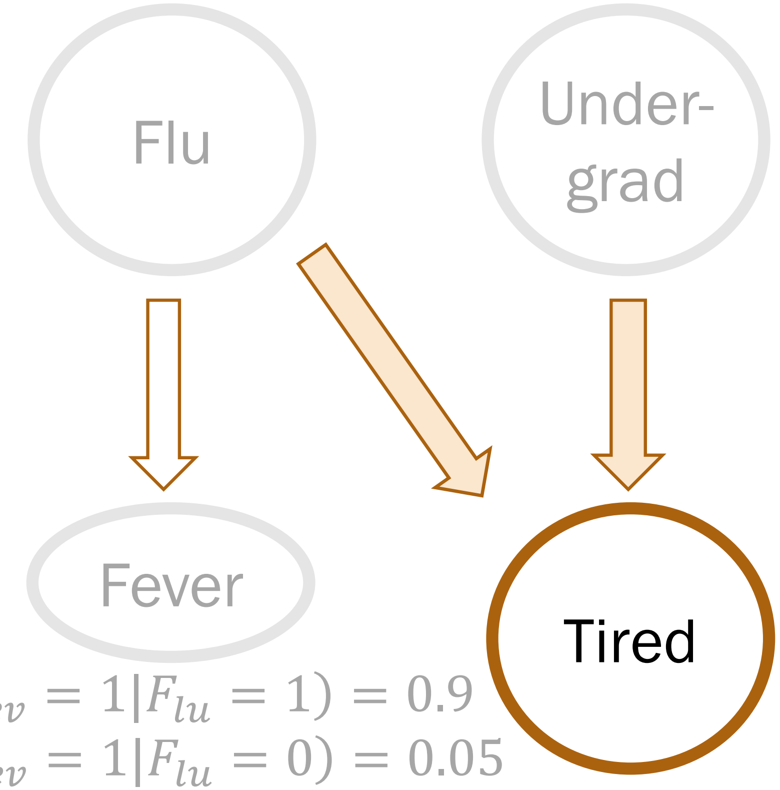
$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

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) # undergraduate

   # 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]
```

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



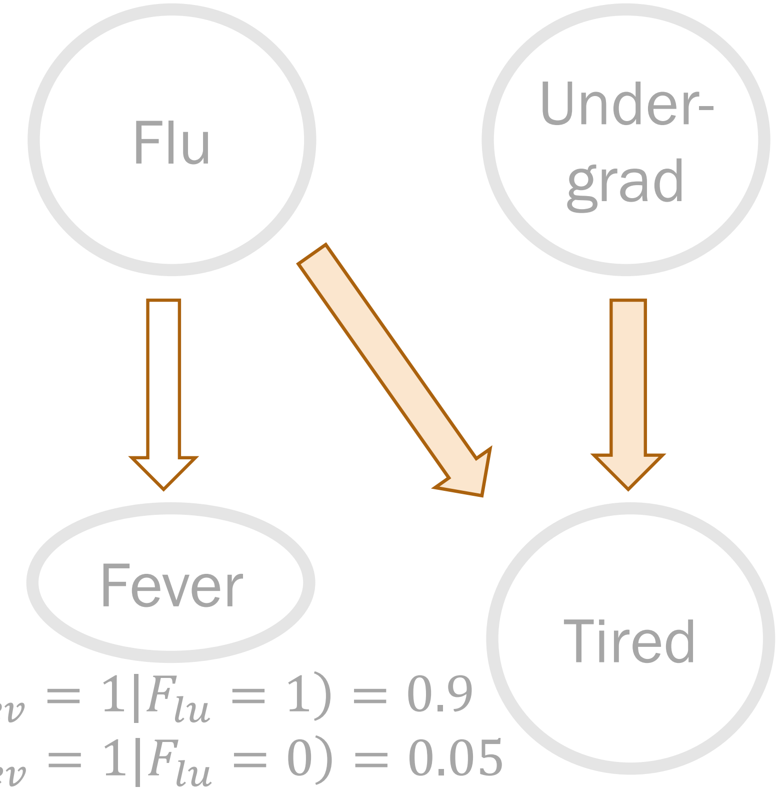
$P(F_{ev} = 1 | F_{lu} = 1) = 0.9$
$P(F_{ev} = 1 | F_{lu} = 0) = 0.05$

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

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) # undergraduate

   # 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]
```

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



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

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

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 =
        reject_inconsistent(samples, observation)
    samples_event =
        reject_inconsistent(samples_observation, event)
    return len(samples_event)/len(samples_observation)
```

[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 =
        reject_inconsistent(samples, observation)
    samples_event =
        reject_inconsistent(samples_observation, event)
    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 $P(F_{lu} = 1 | U = 1, T = 1)$?

```
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 len(samples_event)/len(samples_observation)
```

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

Rejection sampling

Try it yourself!

http://web.stanford.edu/class/cs109/demos/webmd.zip

Rejection sampling

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

[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
```

With enough samples, you can correctly compute:

- Probability estimates
- Conditional probability estimates
- Expectation estimates

Because your samples are a representation of the joint distribution!

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