Table of Contents

15: General Inference

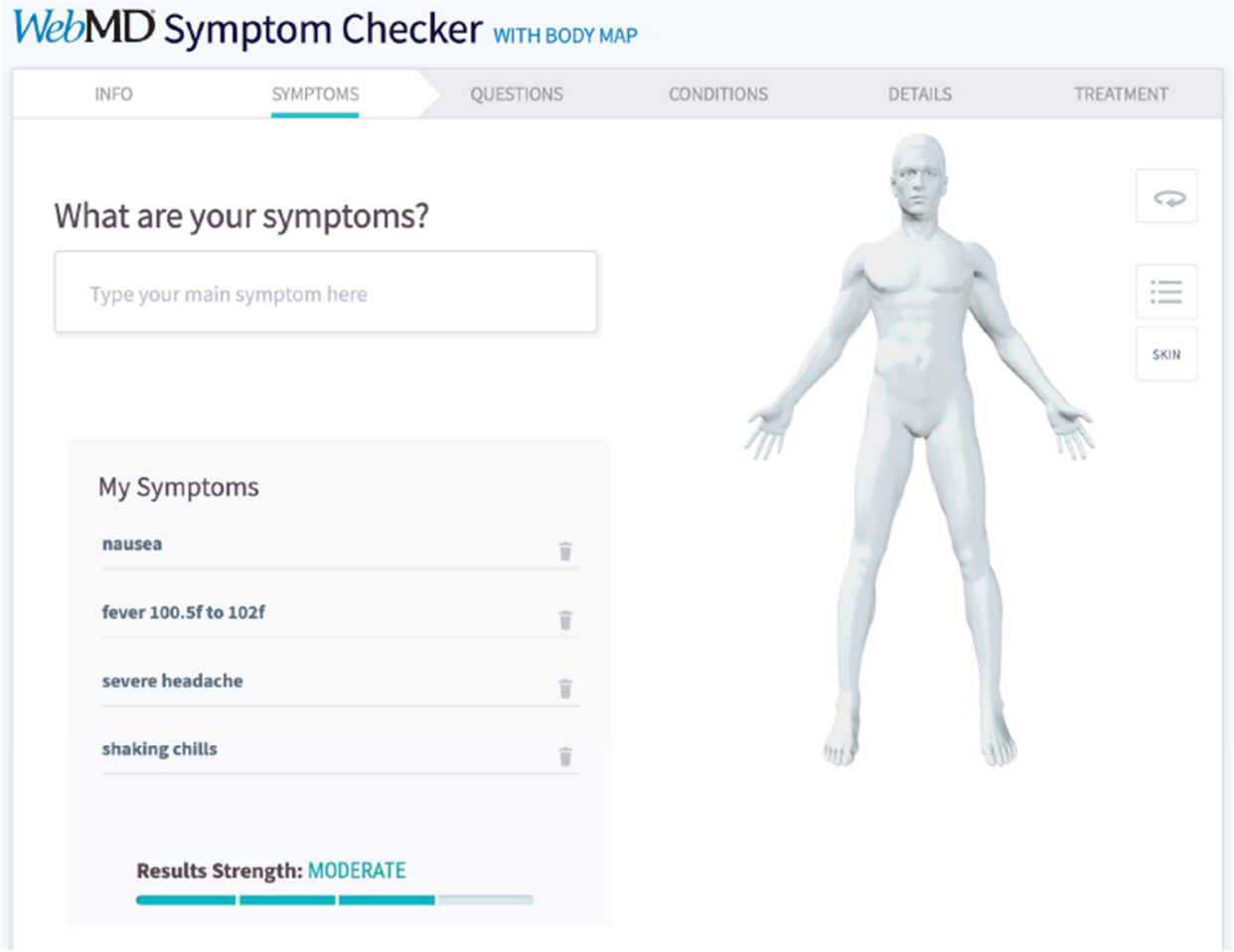
Jerry Cain

February 12, 2024

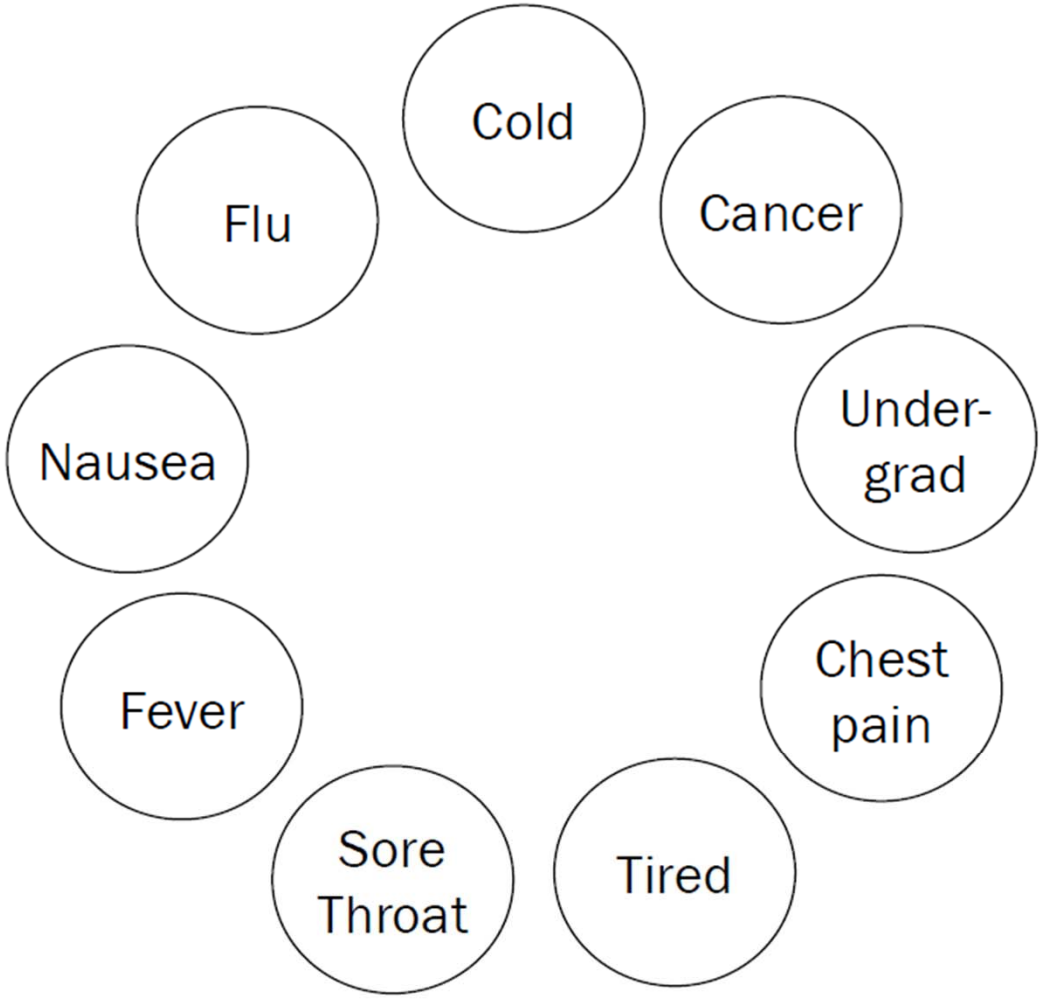
Lecture Discussion on Ed

General Inference: Introduction

Inference


WebMD Symptom Checker WITH BODY MAP
INFO
SYMPTOMS
QUESTIONS
CONDITIONS
DETAILS
TREATMENT
What are your symptoms?
Type your main symptom here
SKIN
My Symptoms
nausea
fever 100.5f to 102f
severe headache
shaking chills
Results Strength: MODERATE

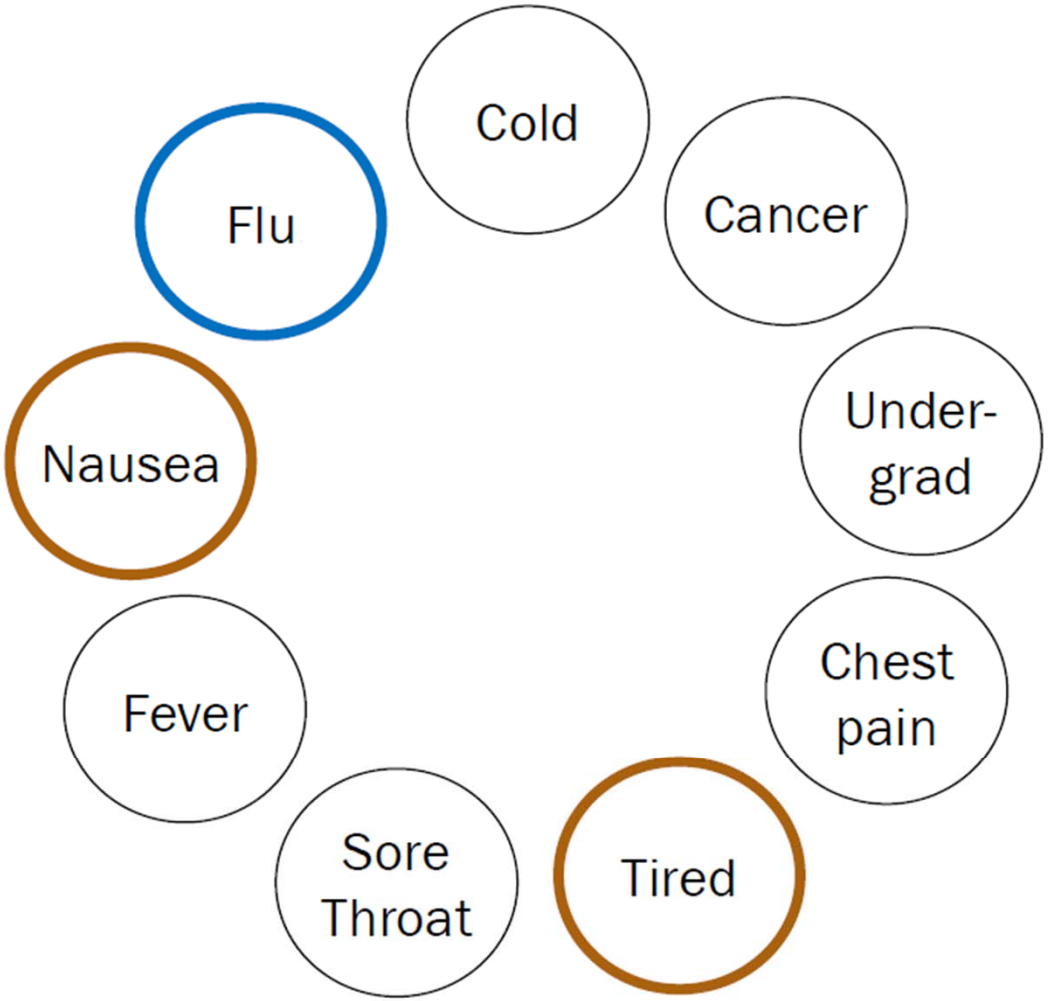
Inference



General inference question:

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

Inference

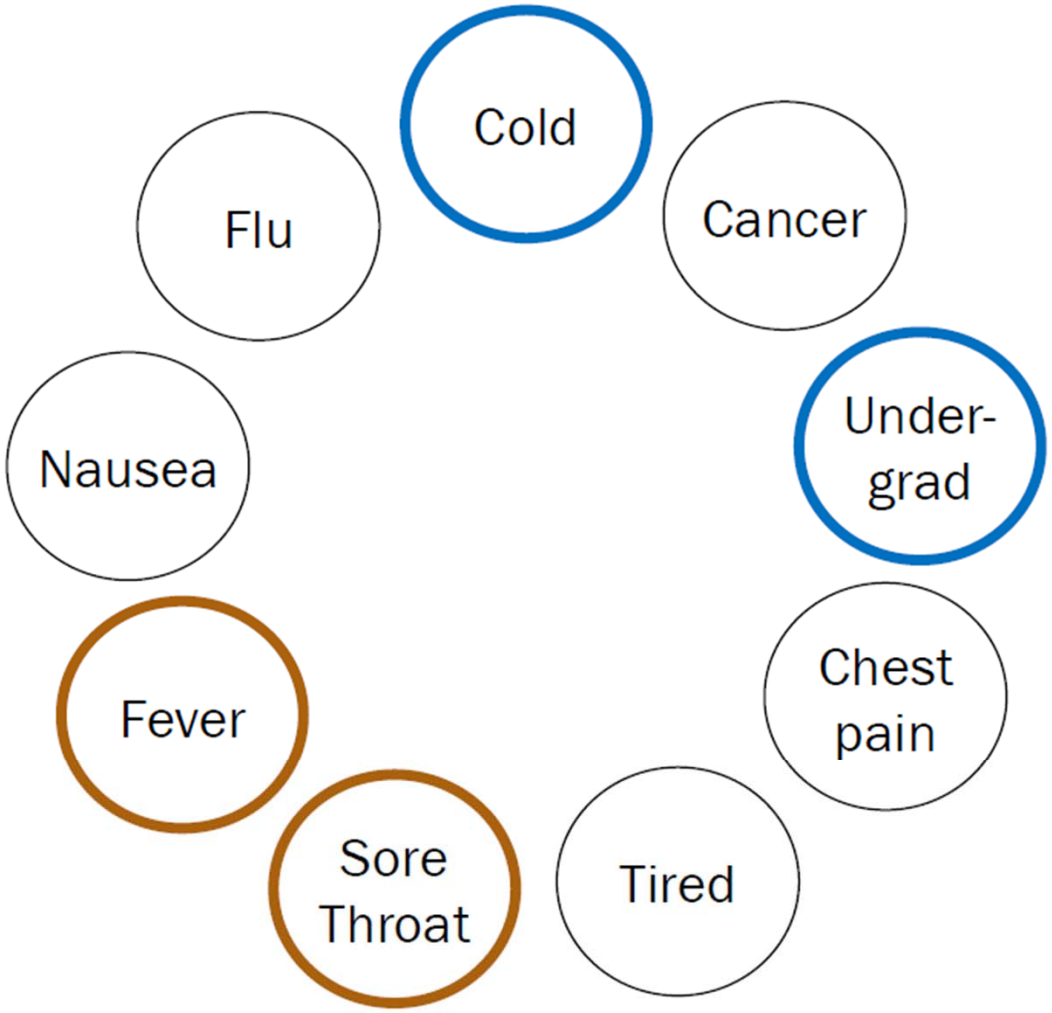


One inference question:

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

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

Inference

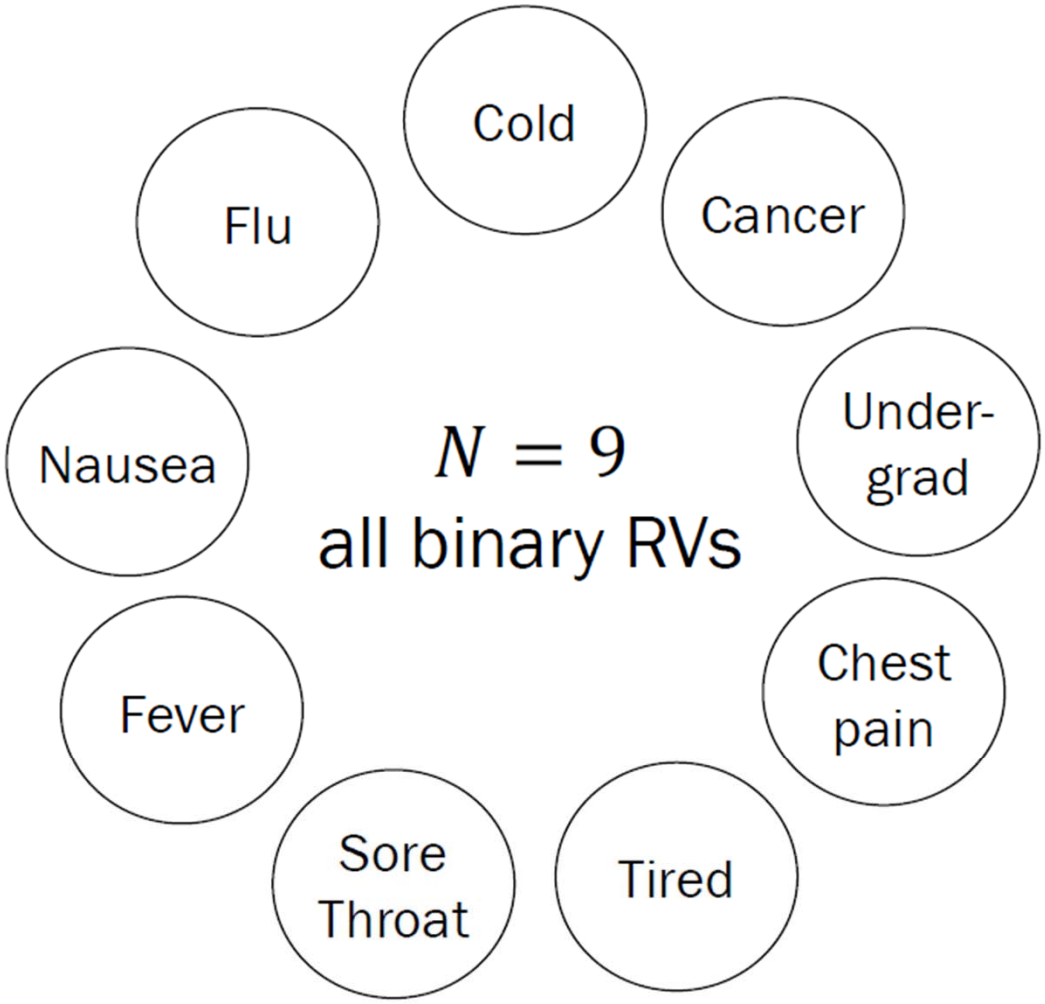


Another inference question:

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

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

Inference

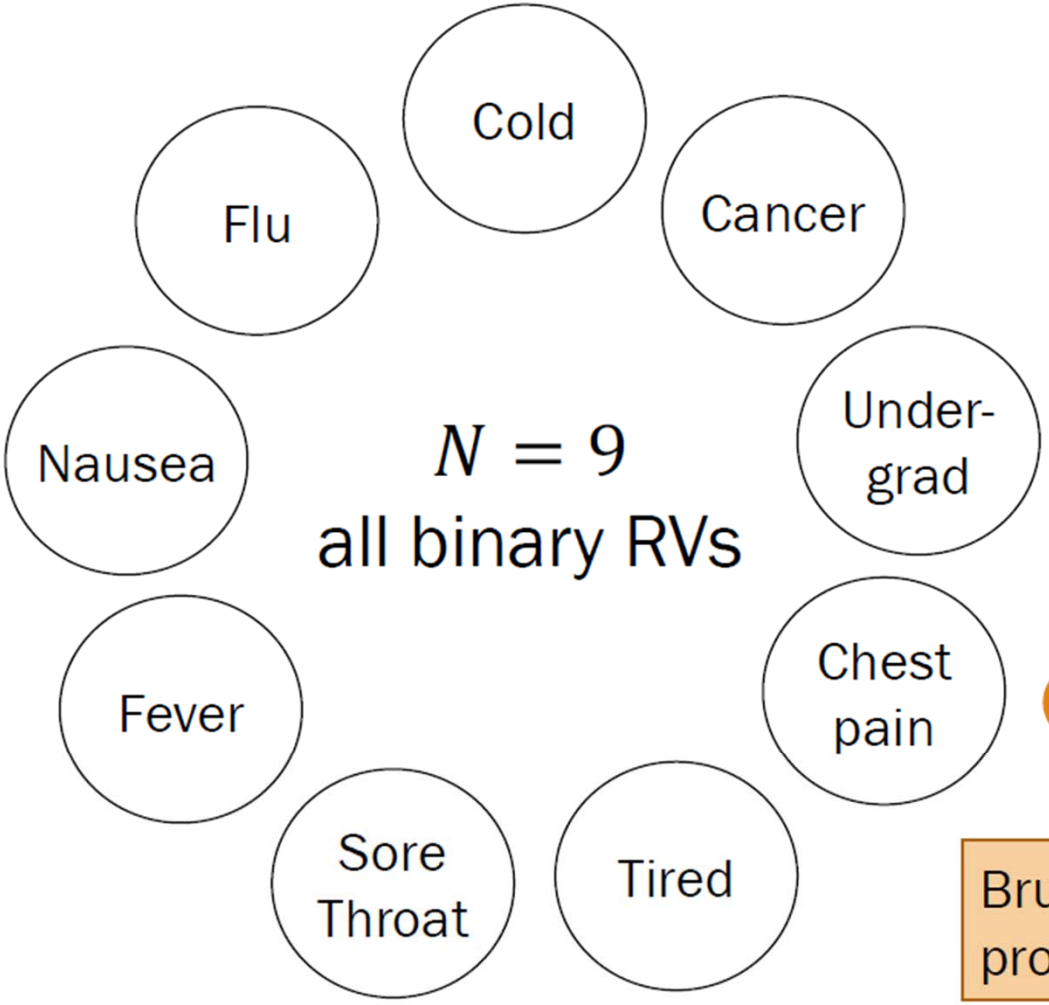


If we know the full **joint distribution**, we can answer all probabilistic inference questions.

What is the size of the joint probability table?

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

Inference



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

What is the size of the joint probability table? why 2^N?

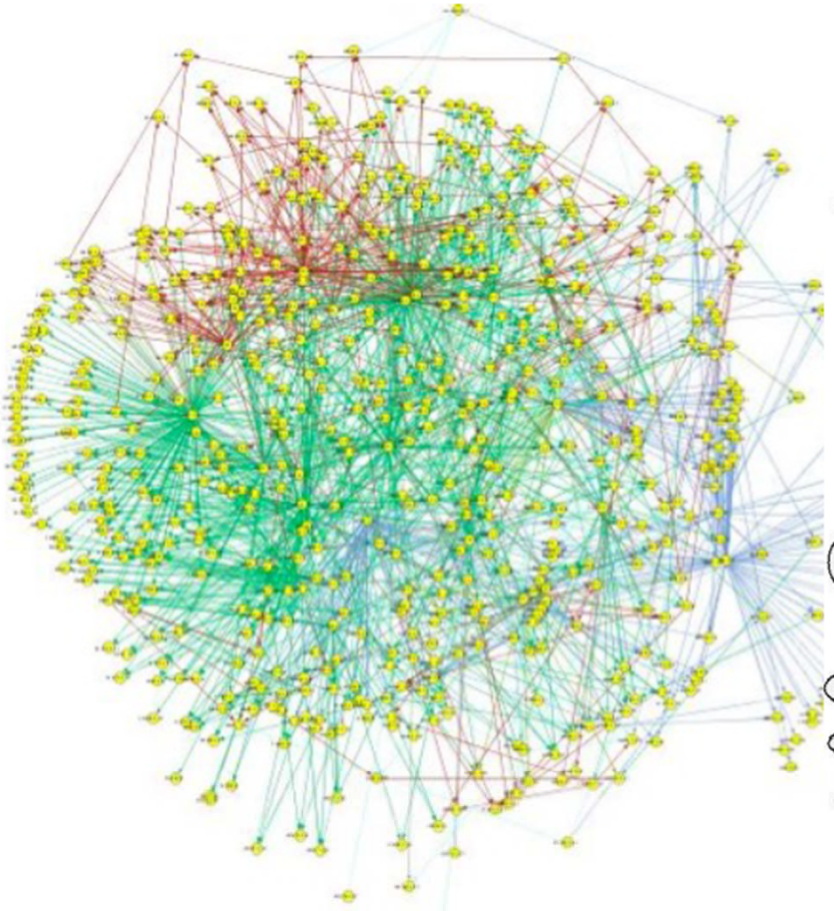
A. 2^{N-1} entries
B. N^N entries
C. 2^N entries
D. None/other/don't know

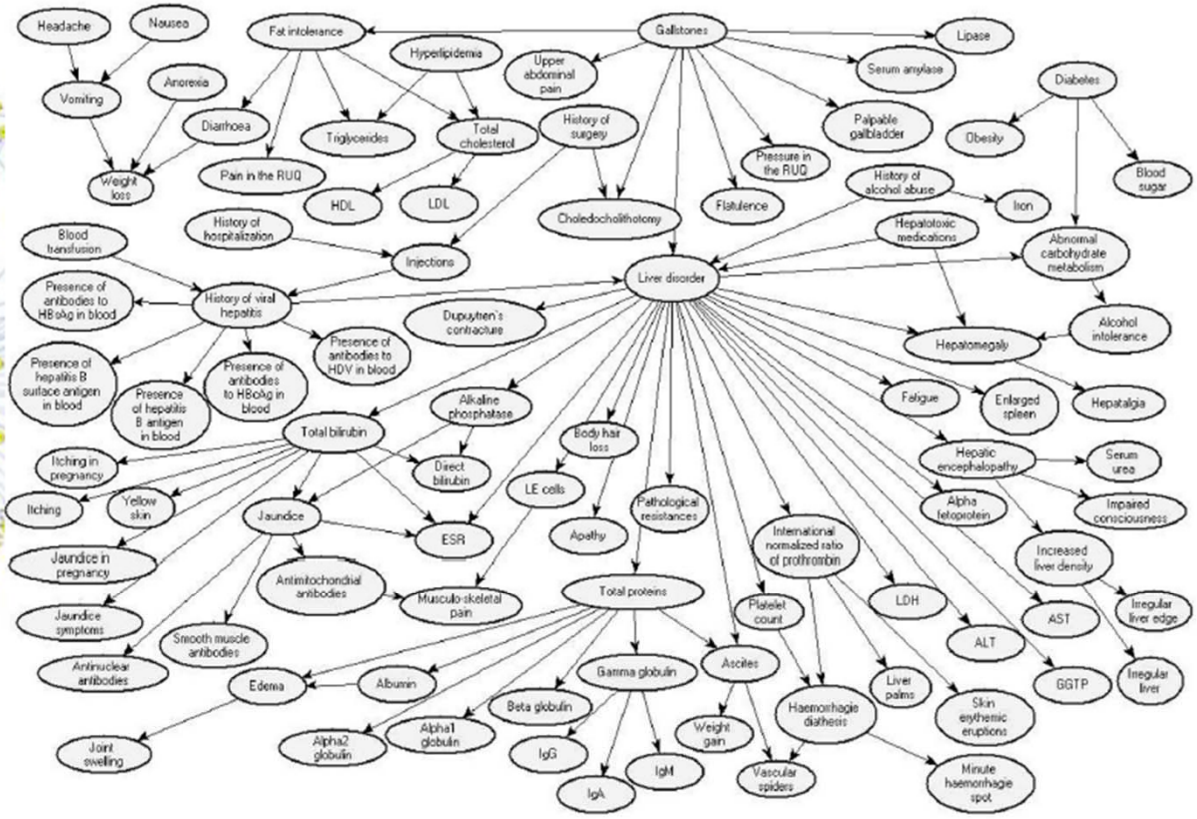
$P(X_1 = \_, X_2 = \_, \ldots, X_N = \_)$
= some value, but each underscore can be filled in in one of two ways.

Brute-force computation of a full joint probability mass function is often intractable.

다루기 힘듦

N can be large…





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

when $n=2$ $P(X_1 = x_1, X_2 = x_2 | Y = y) = P(X_1 = x_1 | Y = y) \cdot P(X_2 = x_2 | Y = y)$

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

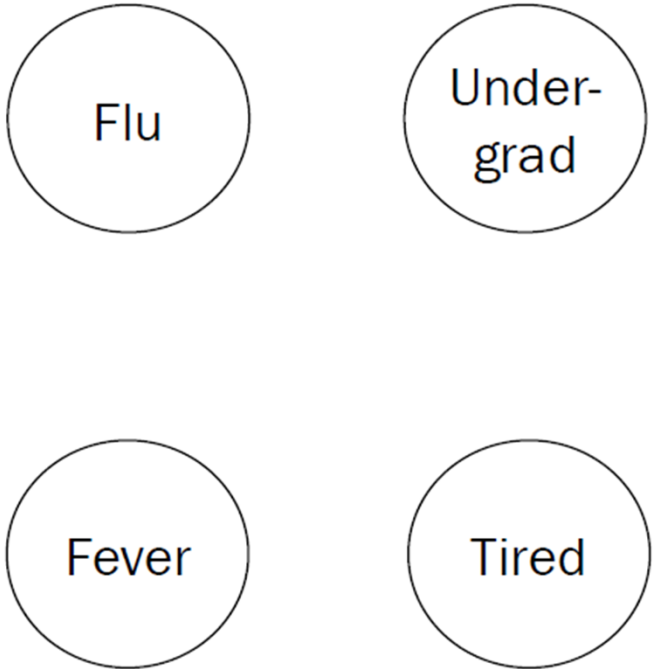
This implies the following (cool to remember for later):

useful when N is large and probabilities are so small that numerical stability is an issue!

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

Bayesian Networks

A simpler WebMD



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

each take on either 0 or 1 independently of each other

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

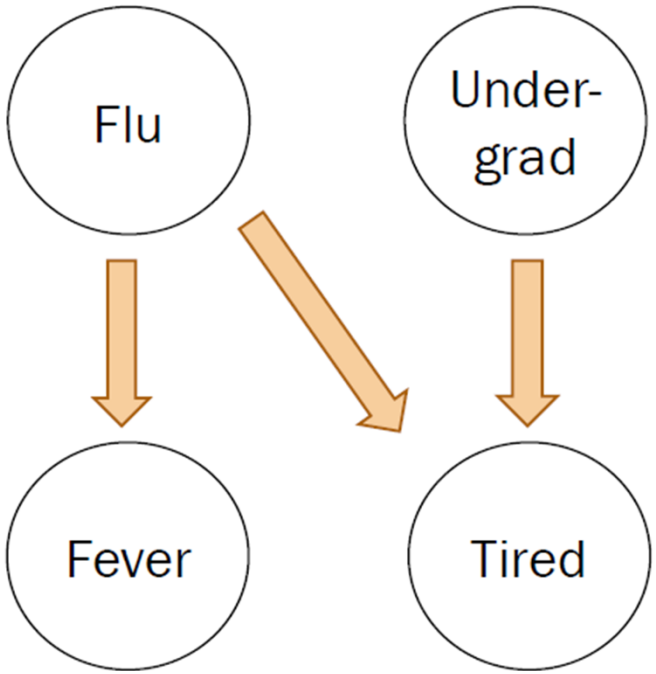
2^4 possibilities

What would an infectious diseases (ID) expert do?

인과관계

Describe the joint distribution using causality!

Constructing a Bayesian Network

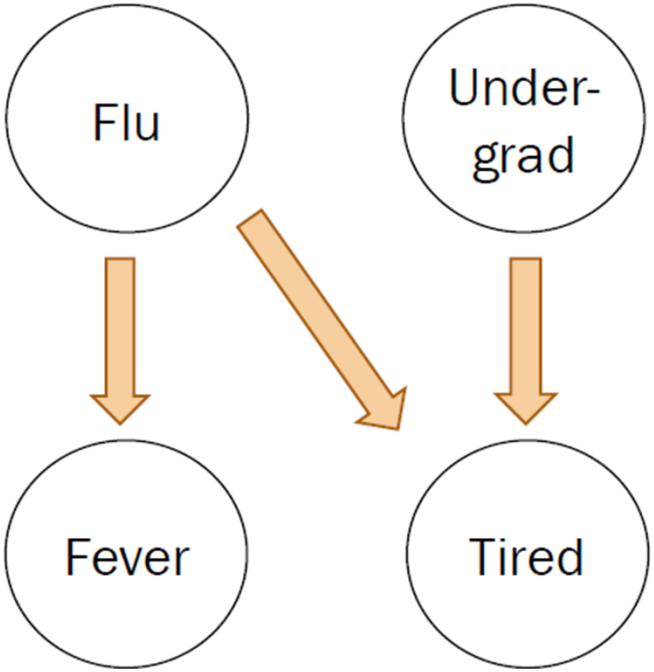


What would an ID expert do?

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

Constructing a Bayesian Network

must be a directed, acyclic graph



In a Bayesian Network,

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

fancy speak that says two variables are conditionally independent of each other if there's no path between them

- Node: random variable
- Directed edge: conditional dependency

in practice, a causal dependency as well

Examples:

doesn't inform Fev

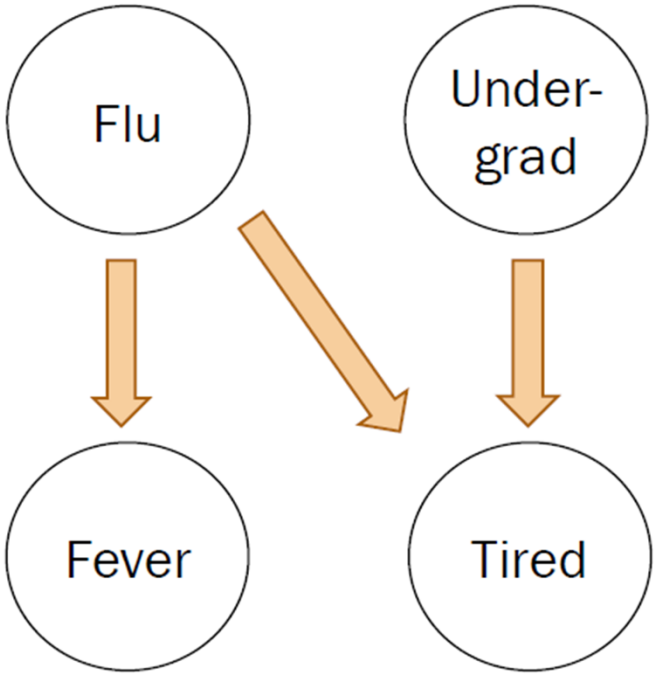
$$P(F_{ev} = 1 | T = 0, F_{lu} = 1) = P(F_{ev} = 1 | F_{lu} = 1)$$

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

no conditional dependencies expressed in the network, so independent

Constructing a Bayesian Network

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



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

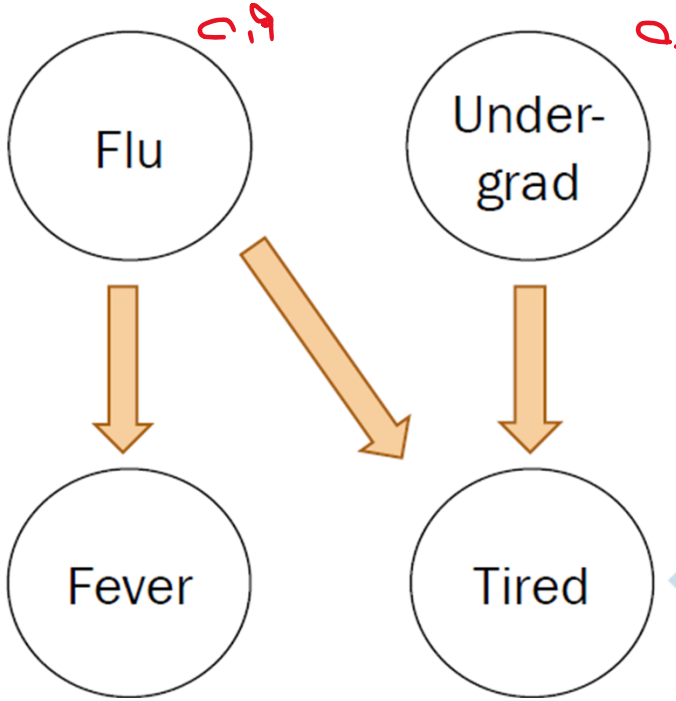
What would an ID expert do?

1. Describe the joint distribution using causality.
2. ✅ Assume conditional independence.
3. Provide $P(\text{values}|\text{parents})$ 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 an ID expert do?

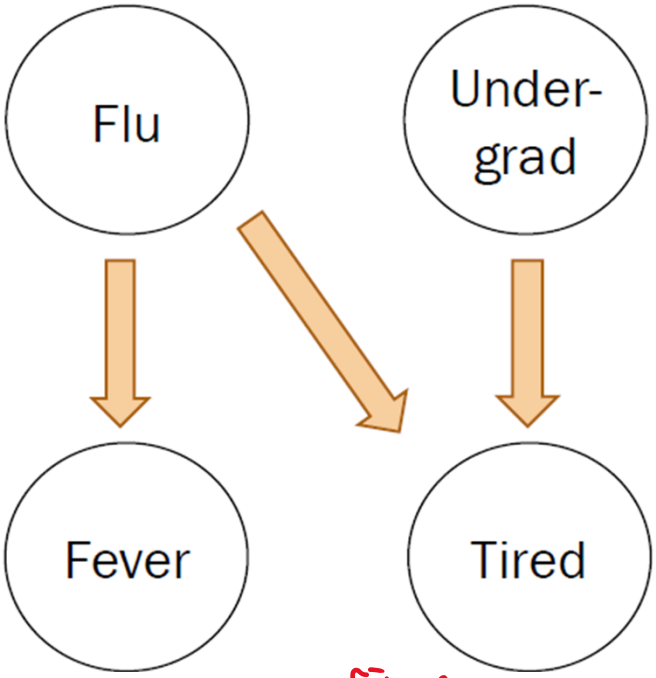
1. Describe the joint distribution using causality.
2. Assume conditional independence.
3. Provide $P(\text{values}|\text{parents})$ for each random variable

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

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



given

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

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

What would a CS109 student do?

1. Populate a Bayesian network by asking an infectious diseases expert
   or
   by using reasonable assumptions

2. Answer inference questions

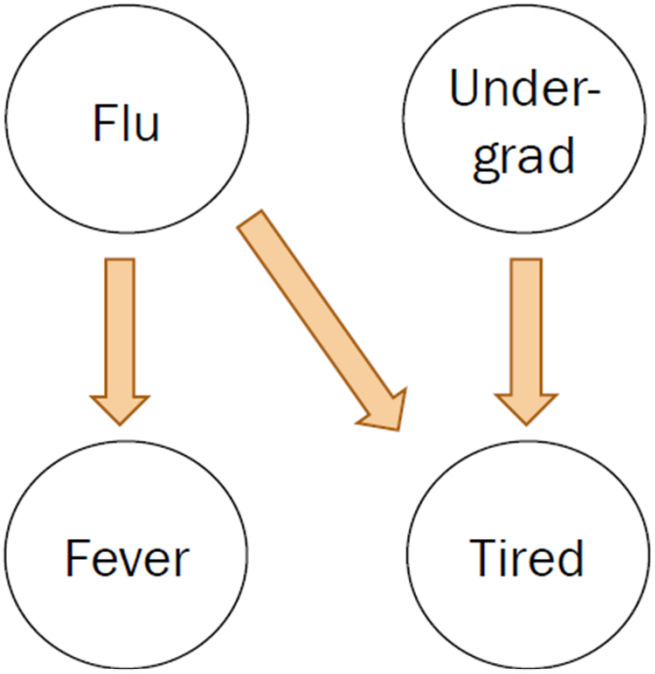
Our focus today

Inference: Math

Inference via math

Chap 4. Chain Rule (aka Product rule)
P(EF) = P(F) P(E|F)

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



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

Compute joint probabilities using chain rule.

$F_{lu} \rightarrow P(F_{lu} = 0) = 0.9$
$U \rightarrow P(U = 1) = 0.8$
$F_{ev} \rightarrow P(F_{ev} = 0 \mid F_{lu} = 0) = 0.95$
$T \rightarrow P(T = 1 \mid F_{lu} = 0, U = 1) = 0.8$

$0.9 \times 0.8 \times 0.95 \times 0.8 = 0.5472$

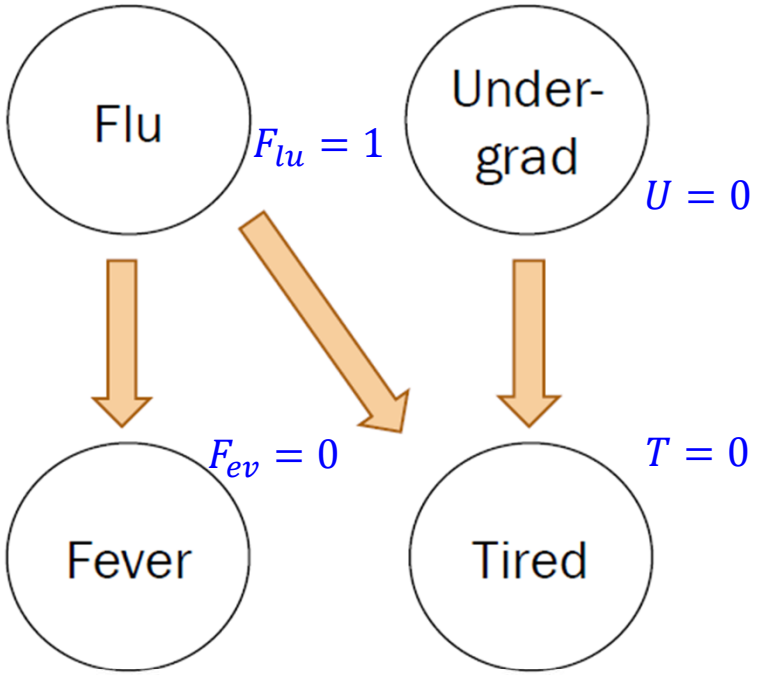
all four variable contribute to joint probability value, but how some contribute is influenced by what they are conditioned on.

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

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

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

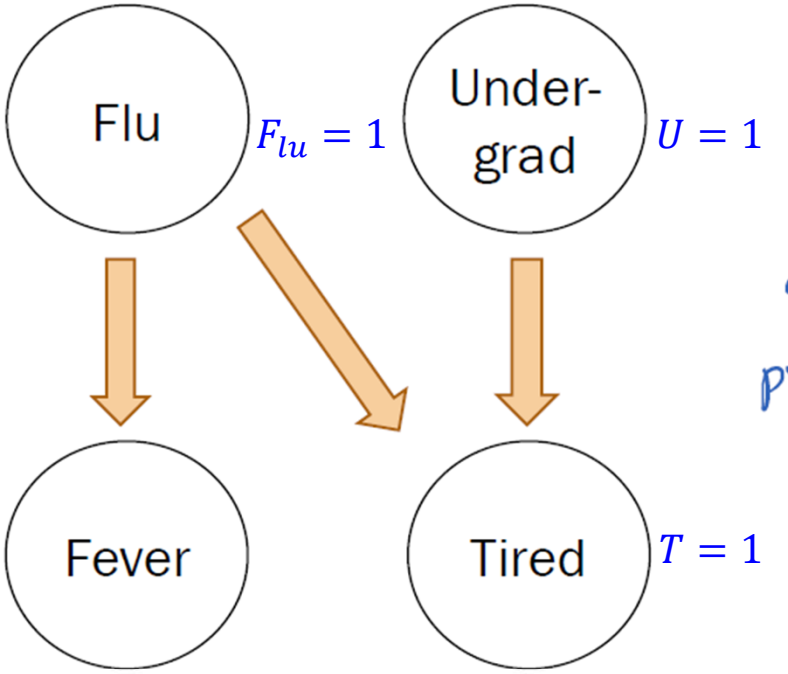
$$\frac{P(F_{lu} = 1, F_{ev} = 0, U = 0, T = 1)}{\sum_x P(F_{lu} = x, F_{ev} = 0, U = 0, T = 1)}$$

denominator results from application of law of total probability with one unknown

$= 0.095$

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

conditional probability that ignores Fev

1. Compute joint probabilities

there are four such probabilities here

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

...

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

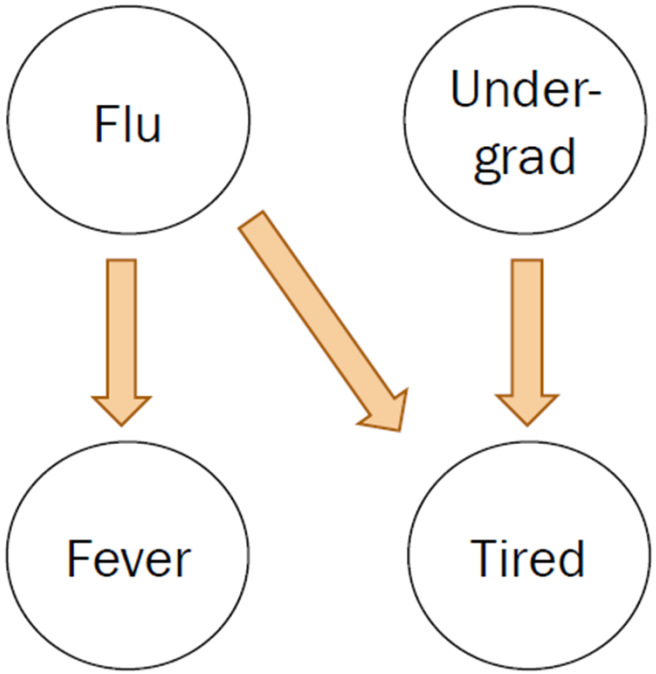
2. Definition of conditional probability

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

$= 0.122$

Inference via math

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



Solving inference questions precisely is possible, but sometimes tedious.

Can we use sampling to do approximate inference?

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

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

Rejection Sampling

정확한 확률밀도 함수를 알기 힘들거나 확률밀도 함수가 주어졌을 때, 해당 함수로부터 sample을 추출하기 어려운 경우 sampling 하는 기본적인 방법

Rejection sampling - Wikipedia
https://en.wikipedia.org/wiki/Rejection_sampling

https://untitledtblog.tistory.com/134

https://untitledtblog.tistory.com/134

Rejection sampling의 기본적인 동작

- 쉽게 샘플을 생성할 수 있는 q에서 샘플들을 생성한 뒤에 이 샘플들의 분포가 p를 따르도록 수정하는 것
- 이를 통해 실제로는 q에서 샘플이 생성되었지만, 그 결과는 p에서 생성된 것처럼 만드는 것이다. 이때 쉽게 샘플을 생성할 수 있도록 임의로 설정한 q를 제안 분포 (proposal distribution)이라고 한다.
- 제안 분포는 uniform distribution, normal distribution 등이 이용될 수 있으며, 가능하면 p와 비슷한 형태의 확률 분포를 사용하는 것이 좋다.

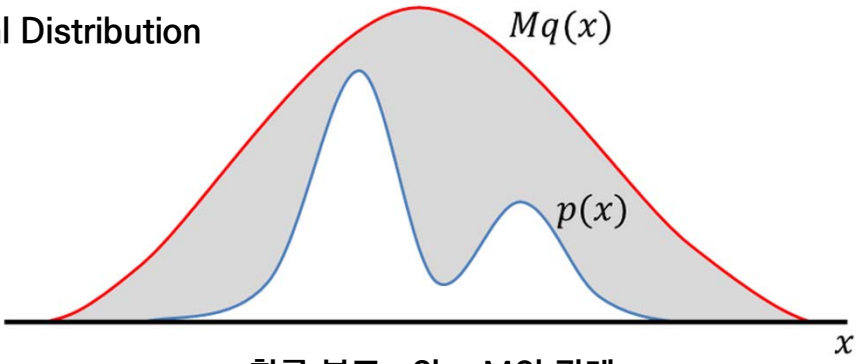
Algorithm 1: Rejection sampling

Input : the number of samples N,
target distribution p,
proposal distribution q,
a given constant M

Output : samples $X = \{x_1, x_2, ..., x_N\}$

$X = \{\}$
while $n < N$ **do**
  $x_0 \sim q(x)$
  $u \sim U(0, 1)$
  if $u < \frac{p(x_0)}{Mq(x_0)}$ **then**
    $X \leftarrow X \cup \{x_0\}$
    $n \leftarrow n + 1$
  end
end

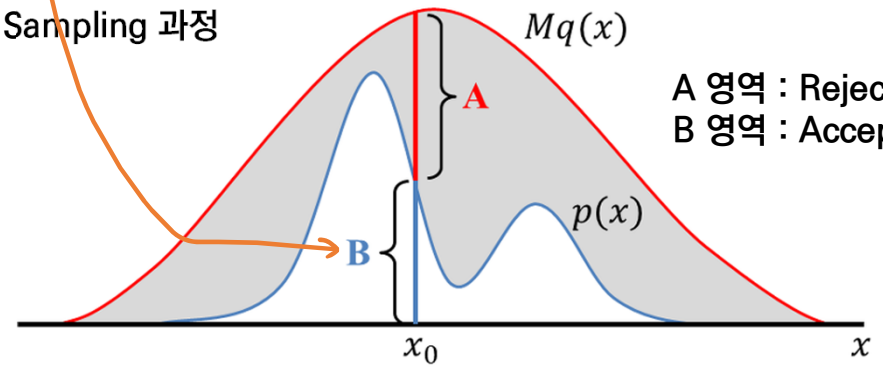
1) Proposal Distribution



확률 분포 p와 q,M의 관계

제안 분포를 설정한 다음에는 상수 M을 모든 x에 대해 $p(x) \leq Mq(x)$이 되도록 설정한다.

2) Sampling 과정



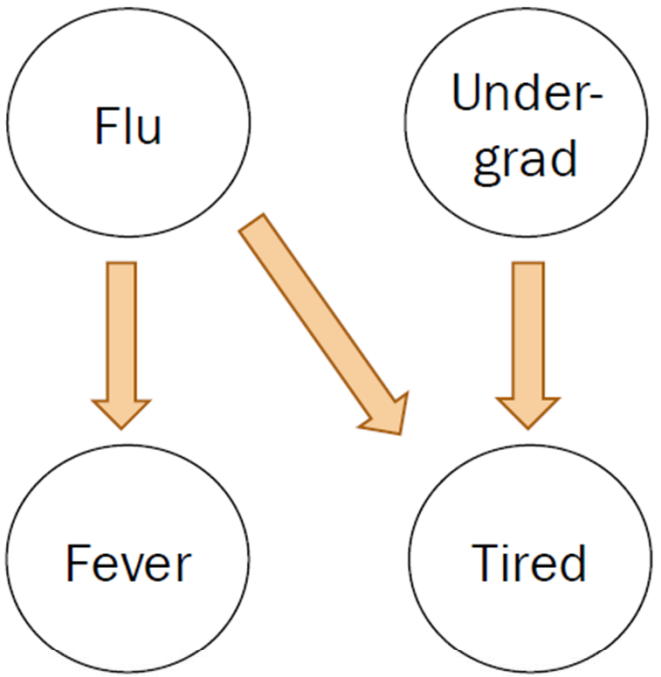
A 영역 : Rejection
B 영역 : Acceptance

확률 분포 q로 부터 샘플 생성 및 Rejection / Acceptance 반복 수용

Rejection sampling algorithm

Step 0:

Require a fully specified Bayesian Network



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

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

Rejection sampling algorithm

this is the observation, or the given information

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)
```

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

Rejection sampling algorithm

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

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

Why would this definition of approximate probability make sense?

this is the frequentist's definition of probability!!

Why would this approximate probability make sense?

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

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

$= \frac{n((F_{lu}=1, U=1, T=1))}{\text{num samples}} \Big/ \frac{n((U=1, T=1))}{\text{num samples}}$

Why would this definition of approximate probability make sense?

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

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

Rejection sampling algorithm

(1) samples=sample_a_ton()

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

(1) samples=sample_a_ton()

```
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?

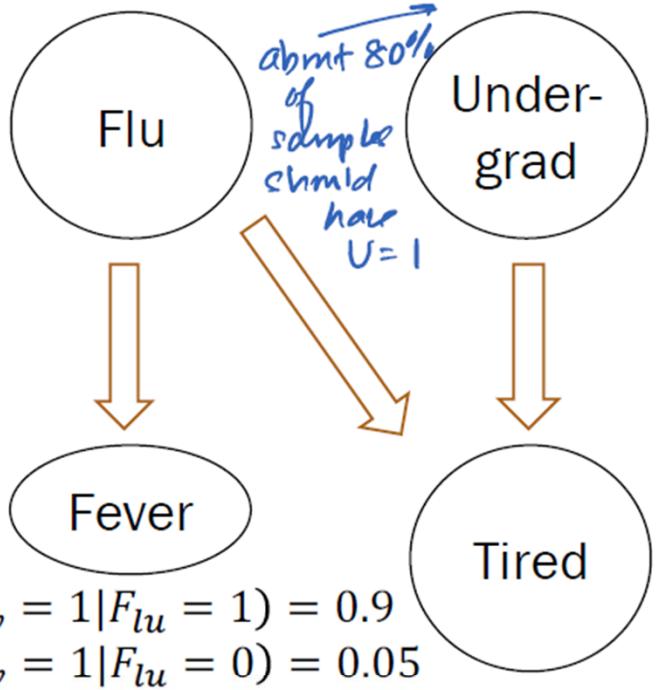
Create a sample using the Bayesian Network!

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]
```

this suggests that about 100% of samples should have $F_{lu} = 0.1$

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



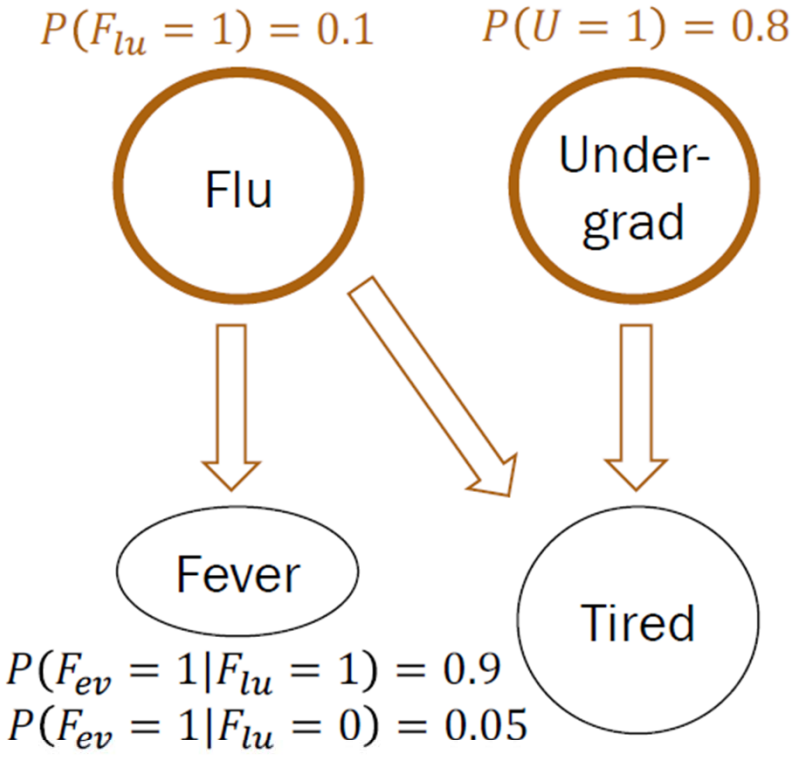
$P(F_{ev} = 1|F_{lu} = 1) = 0.9$
$P(F_{ev} = 1|F_{lu} = 0) = 0.05$

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

Rejection sampling algorithm

(1) samples=sample_a_ton()

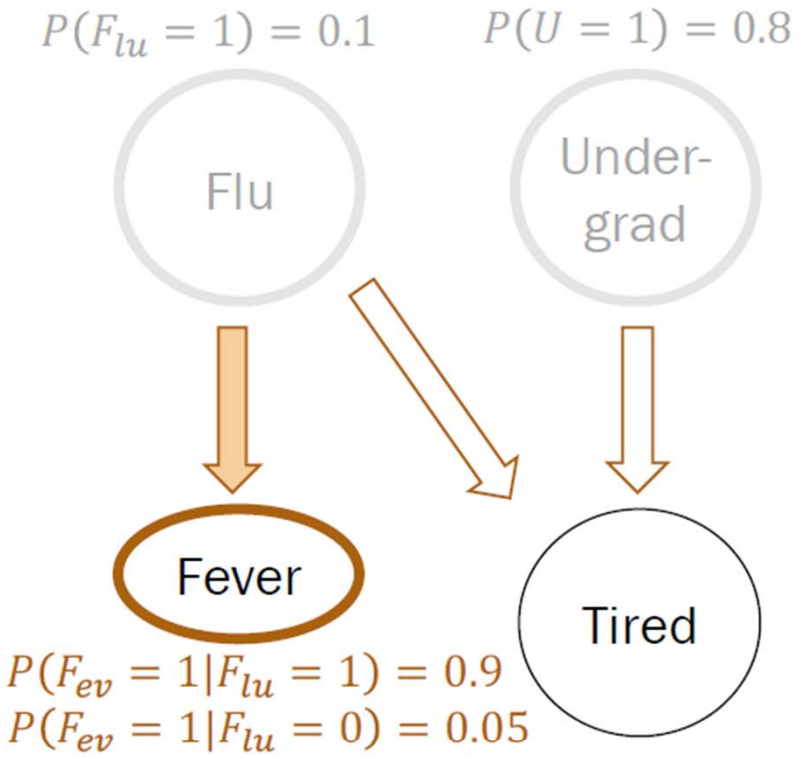
```
# Method: Make Sample
# ------------------
# create a single sample from the joint distributior
# based on the medical "WebMD" Bayesian Network
def make_sample():
    # prior on causal factors
    flu = bernoulli(0.1)
    und = bernoulli(0.8)

    # 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]
```



Rejection sampling algorithm

(1) samples=sample_a_ton()

```
# Method: Make Sample
# ------------------
# create a single sample from the joint distributior
# 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]
```

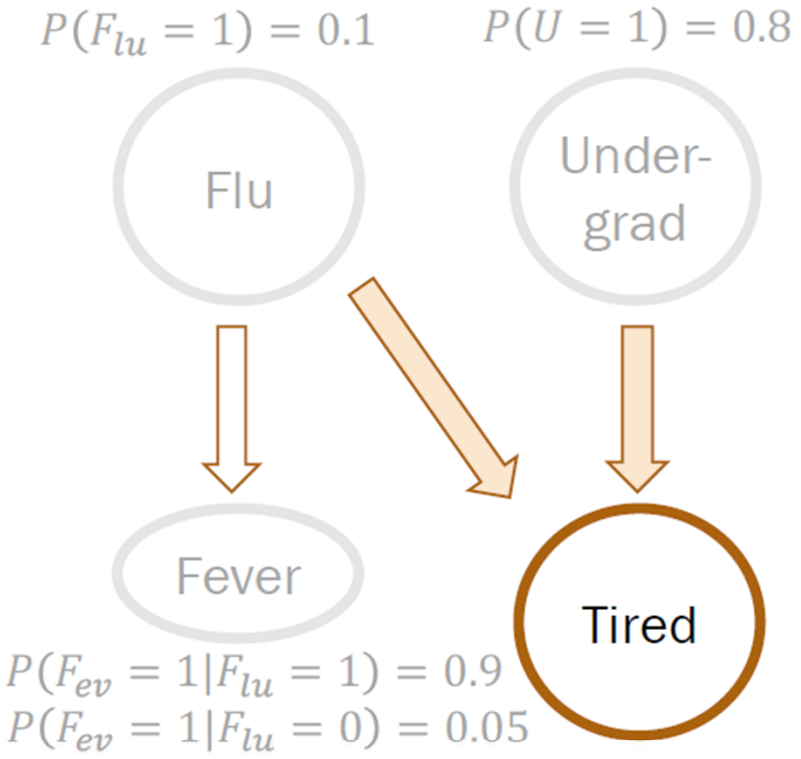


$$P(T = 1|F_{lu} = 0, U = 0) = 0.1$$
$$P(T = 1|F_{lu} = 0, U = 1) = 0.8$$
$$P(T = 1|F_{lu} = 1, U = 0) = 0.9$$
$$P(T = 1|F_{lu} = 1, U = 1) = 1.0$$

Rejection sampling algorithm

(1) samples=sample_a_ton()

```
# Method: Make Sample
# ------------------
# create a single sample from the joint distributior
# 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]
```

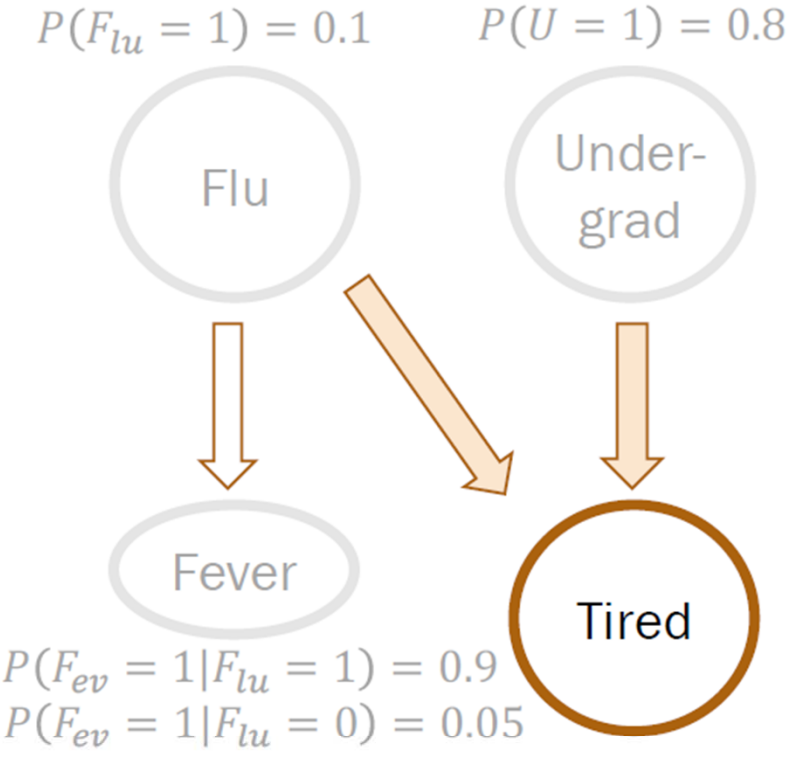


$$P(T = 1|F_{lu} = 0, U = 0) = 0.1$$
$$P(T = 1|F_{lu} = 0, U = 1) = 0.8$$
$$P(T = 1|F_{lu} = 1, U = 0) = 0.9$$
$$P(T = 1|F_{lu} = 1, U = 1) = 1.0$$

Rejection sampling algorithm

(1) samples=sample_a_ton()

```
# Method: Make Sample
# ------------------
# create a single sample from the joint distributior
# 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]
```

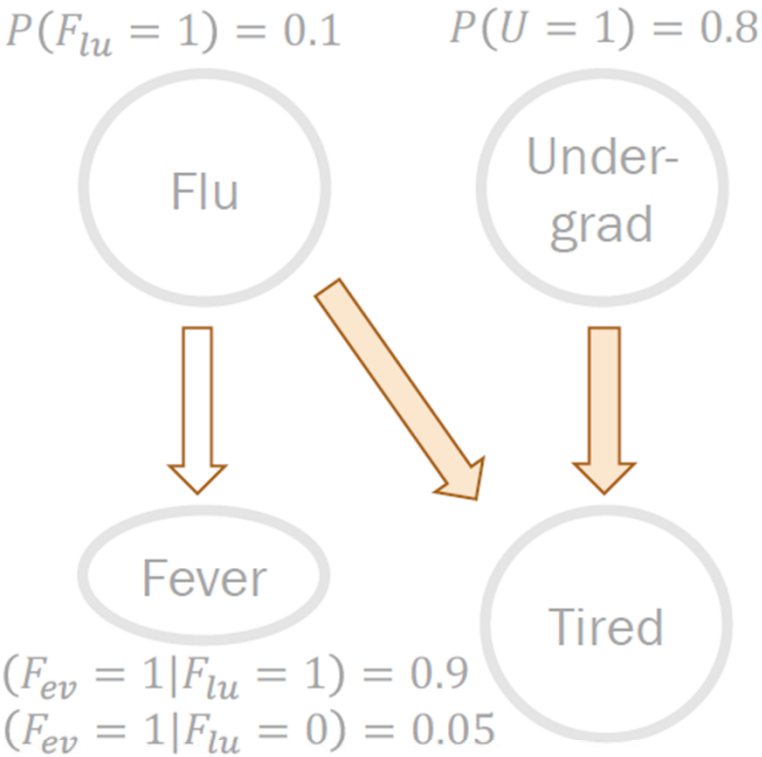


$P(T = 1|F_{lu} = 0, U = 0) = 0.1$
$P(T = 1|F_{lu} = 0, U = 1) = 0.8$
$P(T = 1|F_{lu} = 1, U = 0) = 0.9$
$P(T = 1|F_{lu} = 1, U = 1) = 1.0$

Rejection sampling algorithm

(1) samples=sample_a_ton()

```
# 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]
```



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

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

Rejection sampling algorithm

(2) samples_observation

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)
```

build a new set of samples that retains just those samples consistent with the supplied 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 $P(F_{lu} = 1 | U = 1, T = 1)$?

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

```
# 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
```

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)
```

all these have U=1, T=1

further require S this event to hold as well

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):
    return consistent_samples
```

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

Conditi ation) $= 1)$.

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)
```

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

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

but approximately, even if the approximation is incredibly good

Why? Because your samples represent the joint distribution incredibly well!

```
[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
```