15: General Inference

Jerry Cain
May 3rd, 2024

Lecture Discussion on Ed
Inference
Inference
Inference

What are your symptoms?

Type your main symptom here

My Symptoms

- nausea
- fever 100.5f to 102f
- severe headache
- shaking chills

Results Strength: MODERATE
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

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2024
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

Brute-force computation of a full joint probability mass function is often intractable.
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 $x_1, x_2, \ldots, x_n, y$:

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

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

\[ \log P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n|Y = y) = \sum_{i=1}^{n} \log P(X_i = x_i|Y = y) \]
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 an infectious diseases (ID) expert do?

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.**
In a Bayesian Network, each random variable is \textit{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

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?

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

- $P(F_{ev} = 1|F_{lu} = 1) = 0.9$
- $P(F_{ev} = 1|F_{lu} = 0) = 0.05$
Constructing a Bayesian Network

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?

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

$I.D.$

- $P(F_{lu} = 1) = 0.1$
- $P(U = 1) = 0.8$
Using a Bayes Net

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

- \( P(\text{Flu} = 1) = 0.1 \)
- \( P(U = 1) = 0.8 \)

\[
\begin{align*}
P(\text{Fever} = 1|\text{Flu} = 1) &= 0.9 \\
P(\text{Fever} = 1|\text{Flu} = 0) &= 0.05 \\
P(\text{Tired} = 1|\text{Flu} = 1) &= 0.9 \\
P(\text{Tired} = 1|\text{Flu} = 0) &= 0.1 \\
P(T = 1|\text{Flu} = 0, U = 0) &= 0.1 \\
P(T = 1|\text{Flu} = 0, U = 1) &= 0.8 \\
P(T = 1|\text{Flu} = 1, U = 0) &= 0.9 \\
P(T = 1|\text{Flu} = 1, U = 1) &= 1.0
\end{align*}
\]
Inference: Math
Inference via math

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

1. \[ P(F_{lu} = 0, U = 1, F_{ev} = 0, T = 1) \]

Compute joint probabilities using chain rule.

\[
\begin{align*}
P(F_{ev} = 1|F_{lu} = 1) &= 0.9 \\
P(F_{ev} = 1|F_{lu} = 0) &= 0.05 \\
P(T = 1|F_{lu} = 0, U = 0) &= 0.1 \\
P(T = 1|F_{lu} = 0, U = 1) &= 0.8 \\
P(T = 1|F_{lu} = 1, U = 0) &= 0.9 \\
P(T = 1|F_{lu} = 1, U = 1) &= 1.0
\end{align*}
\]
Inference via math

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

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

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

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

\[ P(F_{ev} = 1|F_{lu} = 1) = 0.9 \quad P(T = 1|F_{lu} = 0, U = 0) = 0.1 \]
\[ P(F_{ev} = 1|F_{lu} = 0) = 0.05 \quad P(T = 1|F_{lu} = 0, U = 1) = 0.8 \]
\[ P(T = 1|F_{lu} = 1, U = 0) = 0.9 \quad P(T = 1|F_{lu} = 1, U = 1) = 1.0 \]

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2024
Inference via math

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

\[
\frac{\sum_y P(F_{lu} = 1, U = 1, F_{ev} = y, T = 1)}{\sum_x \sum_y P(F_{lu} = x, U = 1, F_{ev} = y, T = 1)} = 0.122
\]

\[
P(F_{lu} = 1|F_{lu} = 1) = 0.1
\]

\[
P(U = 1) = 0.8
\]
Solving inference questions precisely is possible, but sometimes tedious.

Can we use sampling to do approximate inference?
Rejection Sampling
Rejection sampling algorithm

Step 0:
Require a fully specified Bayesian Network

Flu

Under-grad

Fever

Tired

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

\[ P(F_{ev} = 1|F_{lu} = 1) = 0.9 \quad P(T = 1|F_{lu} = 0, U = 0) = 0.1 \]

\[ P(F_{ev} = 1|F_{lu} = 0) = 0.05 \quad P(T = 1|F_{lu} = 0, U = 1) = 0.8 \]

\[ P(T = 1|F_{lu} = 1, U = 0) = 0.9 \quad P(T = 1|F_{lu} = 1, U = 1) = 1.0 \]
Rejection sampling algorithm

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

```python
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)
```

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]`
- ...
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 = ...
        # 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)
```

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

```python
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)
```

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

- [flu, und, fev, tir]
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 generate a sample 
\( (F_{lu} = a, U = b, F_{ev} = c, T = d) \) 
that respects the joint probability distributions?

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]
# 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]
```

\[
\begin{align*}
P(F_{lu} = 1) &= 0.1 \\
P(U = 1) &= 0.8 \\
P(F_{ev} = 1|F_{lu} = 1) &= 0.9 \\
P(F_{ev} = 1|F_{lu} = 0) &= 0.05 \\
P(T = 1|F_{lu} = 0, U = 0) &= 0.1 \\
P(T = 1|F_{lu} = 0, U = 1) &= 0.8 \\
P(T = 1|F_{lu} = 1, U = 0) &= 0.9 \\
P(T = 1|F_{lu} = 1, U = 1) &= 1.0
\end{align*}
\]
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]
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]
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]
Rejection sampling algorithm

# Method: Make Sample
# -------------------
# create a single sample from the joint distribution
# based on the medical "WebMD" Bayesian Network

```python
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]
```

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

\[
P(F_{uv} = 1|F_{lu} = 1) = 0.9
\]

\[
P(F_{uv} = 1|F_{lu} = 0) = 0.05
\]

\[
P(T = 1|F_{lu} = 0, U = 0) = 0.1
\]

\[
P(T = 1|F_{lu} = 0, U = 1) = 0.8
\]

\[
P(T = 1|F_{lu} = 1, U = 0) = 0.9
\]

\[
P(T = 1|F_{lu} = 1, U = 1) = 1.0
\]
Rejection sampling algorithm

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

```python
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)
```

[flu, und, fev, tir]

<table>
<thead>
<tr>
<th>Sampling...</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 1, 0, 1]</td>
</tr>
<tr>
<td>[0, 1, 0, 1]</td>
</tr>
<tr>
<td>[0, 1, 0, 1]</td>
</tr>
<tr>
<td>[0, 0, 0, 0]</td>
</tr>
<tr>
<td>[0, 1, 0, 1]</td>
</tr>
<tr>
<td>[0, 1, 1, 1]</td>
</tr>
<tr>
<td>[0, 1, 0, 0]</td>
</tr>
<tr>
<td>[1, 1, 1, 1]</td>
</tr>
<tr>
<td>[0, 0, 1, 1]</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>[0, 1, 0, 1]</td>
</tr>
</tbody>
</table>

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

```python
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 \( P(F_{lu} = 1|U = 1, T = 1) \)?

```python
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_event =
        reject_inconsistent(samples_observation, event)
    return len(samples_event)/len(samples_observation)\n```

Keep only samples that are consistent with the observation $T=1, U=0$.

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

Inference question: $U = 1, T = 1$

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

```python
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)

def reject_inconsistent(samples, outcome):
    Consistent samples:
    $(F_{lu} = x, U = 1, F_{ev} = y, T = 1)$  $(F_{lu} = 1)$
    return consistent_samples
Def \texttt{rejection\_sampling}(event, observation):
    samples = sample\_a\_ton()
    samples\_observation =
        reject\_inconsistent(samples, observation)
    samples\_event =
        reject\_inconsistent(samples\_observation, event)
\textbf{return} len(samples\_event)/len(samples\_observation)

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

Inference question: # samples with \( F_{lu} = 1, U = 1, T = 1 \)
# 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 easily compute:
- Probability estimates
- Conditional probability estimates
- Expectation estimates

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

\[
P(\text{has flu} \mid \text{undergrad and is tired}) = 0.122
\]
Breaking News
Before You Go! CS109 Challenge

Do something cool and creative with probability!

Grand Prize:
All exams replaced with 100%
All Serious Entries:
Extra credit (between 0.5% and 2% added to overall average)

Optional Proposal: Sun. 05/12, 11:59pm
Due: Sun. 06/02, 11:59pm