UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Artificial Intelligence

Bayesian Networks

LESSON 20

prof. Antonino Staiano

M.Sc. In ''Machine Learning e Big Data'' - University Parthenope of Naples

Uncertainty

- Often, an agent has only partial knowledge of the world, and the agent should make the best decision possible even in these situations
- For example, when predicting the weather, the agent has information about today's weather, but there is no way to predict tomorrow's weather with 100% accuracy
- The agent can do better than chance, so the focus is on how to create an agent that makes optimal decisions under limited information and uncertainty

Introduction

- Real-world problems contain **uncertainties due to:**
  - partial observability
  - nondeterminism
  - adversaries
- Example of dental diagnosis using propositional logic

$$Toothache \Rightarrow Cavity$$

  - However inaccurate, not all patients with toothaches have cavities

$$Toothache \Rightarrow Cavity \lor GumProblem \lor Abscess \ldots$$

- To make the rule true, we must add an almost unlimited list of possible problems
- The only way to fix the rule is to make it logically exhaustive

Acting Under Uncertainty

- An agent seeks to make rational decisions by considering the relative importance of various goals and evaluating the likelihood and extent to which these goals can be achieved, however …
- Large domains such as medical diagnosis fail for three main reasons:
  - Laziness: It is too much work to list the complete set of antecedents or consequents needed to ensure an exceptionless rule
  - Theoretical ignorance: Medical science has no complete theory for the domain
  - Practical ignorance: Even if we know all the rules, we might be uncertain about a particular patient because only some necessary tests have been or can be run
- An agent only has a degree of belief in the relevant sentences
- Probability theory
  - tool to deal with degrees of belief in relevant sentences
  - summarizes the uncertainty that comes from our laziness and ignorance

Probability

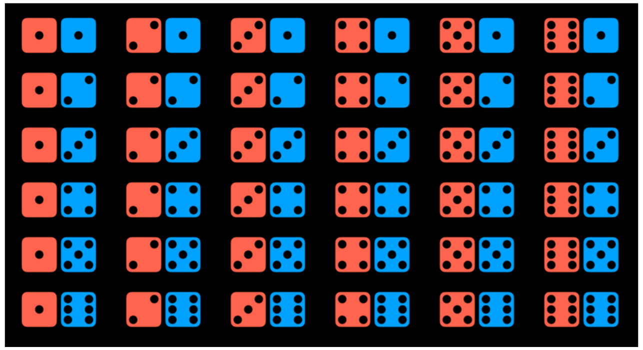
- Uncertainty can be represented as a number of events and the likelihood, or probability, of each of them happening
- Every possible situation can be thought of as a world, represented by the letter $\boldsymbol{\omega}$
- Example
    - Rolling a die can result in six possible worlds:
        - Where the die yields a 1, where it yields a 2, and so on
- The probability of a certain world is denoted as $P(\boldsymbol{\omega})$

Probability Axioms

- $0 \leq P(\omega) \leq 1$: every probability value ranges between 0 and 1
  - 0 is an impossible event
  - 1 is an event that is certain to happen
  - In general, the higher the value, the more likely the event is to happen
- $\sum_{\omega \in \Omega} P(\omega) = 1$
  - The sum of the probabilities of every possible event is equal to 1
- Example
  - The probability of rolling a number R with a die is $P(R) = 1/6$
    - Six possible worlds and each is equally likely to happen

Probability Axioms

- Example: rolling two dice
  - There are 36 possible events equally likely
  - In predicting the sum of the two dice we only have 11 possible events (the sum ranges from 2 to 12)
    - These events do not occur equally as often
  - To get the probability of an event we divide the number of worlds in which it occurs by the number of total possible worlds
    - P(12) = 1/36
    - P(7) = 6/36 = 1/6



Unconditional probability

- Is the degree of belief in a proposition in the absence of any other *evidence*
- All the questions that we have asked so far were questions of unconditional probability
  - The result of rolling a die is not dependent on previous events

Conditional probability

- Is the degree of belief in a proposition given some evidence that has already been revealed
- An agent can use partial information to make consistent guesses about the future
  - To use this information, which affects the probability that the event occurs in the future, it relies on conditional probability
- Conditional probability id denoted as P(a | b)
  - The probability of event a occurring given that we know event b to have occurred, in other words, the probability of **a** given **b**

Conditional Probability

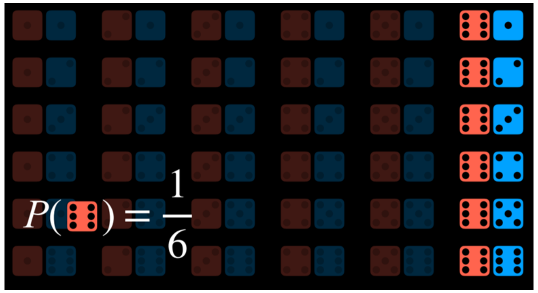
- Now we can ask questions like
  - "What is the probability of rain today given that it rained yesterday"
    - P(*rain today* | *rain yesterday*)
  - "What is the probability of the patient having the disease given their test results"
    - P(*disease* | *test results*)

Conditional Probability

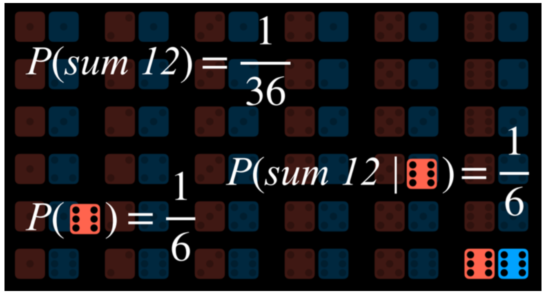
- From a mathematical point of view
- $P(a\,|b) = \frac{P(a \wedge b)}{P(b)}$ (1)
  - Intuitively, we are interested in situations where the events a and b both happen, $P(a \wedge b)$, but only from worlds where we know b to be happened, $P(b)$
- (1) is equivalent to
  - $P(a \wedge b) = P(b)\,P(a\,|b)$
  - $P(a \wedge b) = P(a)\,P(b\,|a)$

Example

- P(*sum 12* | *roll 6 on one die*)
  - To calculate this, we first restrict our worlds to the ones where the value of the first die is 6:



  - Now we ask how many times the event a (the sum being 12) occurs in the worlds where the first die rolled 6 (dividing by P(b), or the probability of the first die yielding 6)



Random Variables

- In probability theory, a random variable is a variable with a range of possible values it can take
  - E.g., to represent the possible outcomes of rolling a die, we can define a random variable *Roll* that can take the values {*1, 2, 3, 4, 5, 6*}
  - E.g., to represent the status of a flight, we can define a variable *Flight* that can take the values {*on time, delayed, canceled*}
- We are interested in the probability with which each value occurs
  - By using a probability distribution

Probability Distribution

- P(Flight = on time) = 0.6
- P(Flight = delayed) = 0.3
- P(Flight = canceled) = 0.1

- A probability distribution can be represented as **P**(*Flight*) = <0.6, 0.3, 0.1>
  - To interpret this notation, the values have a set order, i.e., *on time*, *delayed*, *canceled*

Independence

- Independence is the knowledge that the occurrence of one event does not affect the probability of the other event
  - For example, when rolling two dice, the result of each die is independent of the other
- This is opposed to dependent events, like clouds in the morning and rain in the afternoon
  - If it is cloudy in the morning, it is more likely that it will rain in the afternoon, so these events are dependent
- Independence is defined mathematically:
  - events *a* and *b* are independent if and only if the probability of *a* and *b* is
    - $P(a \land b) = P(a)P(b)$

Bayes' Rule

- Bayes' rule is commonly used in probability theory to compute conditional probability

$$P(b|a) = \frac{P(b)\,P(a|b)}{P(a)}$$

- Example
  - To compute the probability of it raining in the afternoon if there are clouds in the morning, P(rain | clouds), we start with the following information
    - 80% of rainy afternoons start with cloudy mornings, or P(*clouds* | *rain*)
    - 40% of days have cloudy mornings, or P(*clouds*)
    - 10% of days have rainy afternoons, or P(*rain*)
  - Applying Bayes' rule, we compute (0.1)(0.8)/(0.4) = 0.2
    - That is, the probability that it rains in the afternoon given that it was cloudy in the morning is 20%

Joint Probability

- Joint probability is the likelihood of multiple events all occurring
  - Let us consider the following example, the probabilities of clouds in the morning and rain in the afternoon

C = ***cloud***	**C =** ***¬cloud***
0.4	0.6

R = ***rain***	**R =** ***¬rain***
0.1	0.9

  - Looking at these data, we can't say whether clouds in the morning are related to the likelihood of rain in the afternoon
    - To be able to do so, we need to look at the joint probabilities of all the possible outcomes of the two variables
    - We can represent this in a table as follows:

	R = ***rain***	**R =** ***¬rain***
C = *cloud*	0.08	0.32
C = *¬cloud*	0.02	0.58

    - Now we can know information about the co-occurrence of the events
      - For example, we know that the probability of a certain day having clouds in the morning and rain in the afternoon is 0.08
      - The probability of no clouds in the morning and no rain in the afternoon is 0.58

Joint Probability

	R = *rain*	R = *¬rain*
C = *cloud*	0.08	0.32
C = *¬cloud*	0.02	0.58

- Using joint probabilities, we can deduce the conditional probability
  - For example, if we are interested in the probability distribution of clouds in the morning given rain in the afternoon: P(*cloud* | *rain*) = P(*cloud*, *rain*)/P(*rain*)
  - In the last equation, it is possible to view P(rain) as some constant by which P(C, rain) is multiplied
    - Thus, we can rewrite P(C, rain)/P(rain) = **α**P(C, rain), or **α**<0.08, 0.02>
  - Factoring out **α** leaves us with the proportions of the probabilities of the possible values of C given that there is rain in the afternoon
  - Namely, if there is rain in the afternoon, the proportion of the probabilities of clouds in the morning and no clouds in the morning is 0.08:0.02
    - Note that 0.08 and 0.02 don't sum up to 1; however, since this is the probability distribution for the random variable C, we know that they should sum up to 1
    - Therefore, we need to normalize the values by computing **α** such that **α**0.08 + **α**0.02 = 1
    - Finally, we can say that P(C | rain) = <0.8, 0.2>

Probability Rules

- Negation: $P(\neg a) = 1 - P(a)$
  - Because the sum of the probabilities of all the possible worlds is 1, and the complementary literals a and ¬a include all the possible worlds
- Inclusion-Exclusion: $P(a \lor b) = P(a) + P(b) - P(a \land b)$
  - the worlds in which a or b are true are equal to all the worlds where a is true, plus the worlds where b is true
  - However, in this case, some worlds are counted twice (the worlds where both a and b are true)
    - To get rid of this overlap, we subtract once the worlds where both a and b are true (since they were counted twice)

Probability rules

- Marginalization: P(a) = P(a, b) + P(a, ¬b)
  - The idea here is that b and ¬b are disjoint probabilities, i.e., the probability of b and ¬b occurring at the same time is 0
  - We also know b and ¬b sum up to 1
  - When a happens, b can either happen or not
    - Taking the probability of both a and b happening in addition to the probability of a and ¬b, we end up with simply the probability of a
- Marginalization can be expressed for random variables in the following way

$$P(X = xi) = \sum_{j} P(X = xi, Y = yj)$$

- Example
  - P(C = cloud) = P(C = cloud, R = rain) + P(C = cloud, R = ¬rain) = 0.08 + 0.32 = 0.4

	R = ***rain***	**R =** ***¬rain***
C = *cloud*	0.08	0.32
C = *¬cloud*	0.02	0.58

Probability Rules

- Conditioning: P(a) = P(a | b)P(b) + P(a | ¬b)P(¬b)
  - This is a similar idea to marginalization

$$P(X = xi) = \sum_j P(X = xi, | Y = yj)P(Y = yj)$$

Bayesian Networks

Representing Knowledge in an Uncertain Domain

- Bayesian Networks
  - Represents dependencies among (random) variables
- A simple directed graph in which each node is annotated with quantitative probability information
  - Syntax
    - A set of nodes, one per variable
    - A directed, acyclic graph (a link means "directly influences")
      - Arrow from X to Y means X is parent of Y
    - Each node X_i has a conditional distribution given its parents $\mathbf{P}(X_i | Parents(X_i))$

Representing Knowledge in an Uncertain Domain

- Semantics
  - The full joint distribution is the product of the node conditional distributions

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1}) = \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i))$$

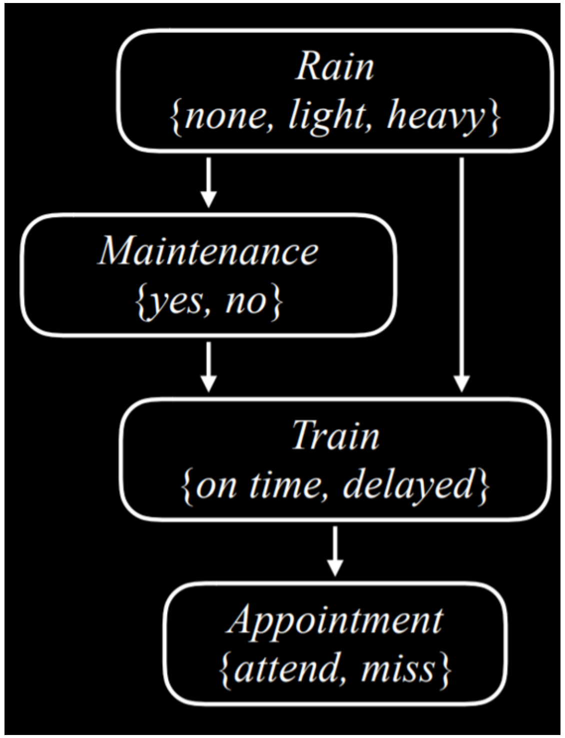
Example

- Getting to an appointment on time
  - Rain is the root node in this BN
    - The probability distribution is not reliant on any prior event
    - It's a random variable that can take the values (none, light, heavy) with a probability distribution

none	*light*	*heavy*
0.7	0.2	0.1

  - Maintenance encodes whether there is train track maintenance, values (yes, no)
    - Its probability distribution is affected by Rain (Rain is its parent node)

R	*yes*	*no*
none	0.4	0.6
light	0.2	0.8
heavy	0.1	0.9



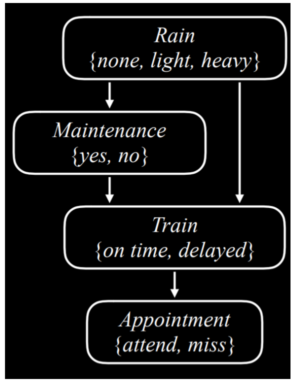
Example (cont.)

- Train encodes whether the train is on time or delayed, values (on time, delayed)
    - It is affected by both Rain and Maintenance

R	M	*on time*	*delayed*
none	yes	0.8	0.2
none	no	0.9	0.1
light	yes	0.6	0.4
light	no	0.7	0.3
heavy	yes	0.4	0.6
heavy	no	0.5	0.5

- Appointment represents whether one attends his appointment, values (attend, miss)

T	*attend*	*miss*
on time	0.9	0.1
delayed	0.6	0.4



- Note that the only parent of Appointment is Train
    - Parents include only direct relations
    - For example, if the train arrived on time, there could be heavy rain and track maintenance, but that would not affect whether we made our appointment

Example (cont.)

- Let's pretend to find the probability of missing the meeting when the train was delayed on a day with no maintenance and light rain, or P(light, no, delayed, miss):
  - P(light)P(no | light)P(delayed | light, no)P(miss | delayed)
  - The value of each of the individual probabilities can be found in the probability distributions, and then these values are multiplied to produce the probability we're looking for, i.e., 0.0192

none	***light***	***heavy***
0.7	0.2	0.1

R	***yes***	***no***
none	0.4	0.6
light	0.2	0.8
heavy	0.1	0.9

R	**M**	***on time***	***delayed***
none	yes	0.8	0.2
none	no	0.9	0.1
light	yes	0.6	0.4
light	no	0.7	0.3
heavy	yes	0.4	0.6
heavy	no	0.5	0.5

T	***attend***	***miss***
on time	0.9	0.1
delayed	0.6	0.4

Inference

- We can infer new information from probabilities
  - Not information for certain but probability distributions for some values
- Property of inference
  - Query X
    - the variable for which we want to compute the probability distribution
  - Evidence variable E
    - One or more variables that have been observed for event e
  - Hidden variable Y
    - variables that aren't the query and also haven't been observed
  - The goal
    - Compute $P(X|e)$

Inference example

- Probability distribution of *Appointment* variable
  - Given the evidence that there is *light rain* and *no track maintenance*
- *P(Appointment | light, no)*
  - We can express the possible values of *Appointment* as a proportion, rewriting
    - *P(Appointment | light, no)* as αP(Appointment, light, no)
  - Through marginalization we get

P(Appointment, light, no) = ***α***[P(Appointment, light, no, delayed) + P(Appointment, light, no, on time)]

Inference by Enumeration

- A process of finding the probability distribution of variable X given observed evidence and some hidden variables

$$P(X|\mathrm{e}) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y)$$

- X is the query variable
- e is the observed evidence
- y for all the values of the hidden variables
- α normalization value (probabilities adding up to 1)

BN in Python

- First, we create the nodes and provide a probability distribution for each one

```
from pomegranate import *
# Rain node has no parents
rain = Node(DiscreteDistribution({
"none": 0.7,
"light": 0.2,
"heavy": 0.1
}), name="rain")
```

BN in Python

```
# Train node is conditional on rain and maintenance
train = Node(ConditionalProbabilityTable([
["none", "yes", "on time", 0.8],
["none", "yes", "delayed", 0.2],
["none", "no", "on time", 0.9],
["none", "no", "delayed", 0.1],
["light", "yes", "on time", 0.6],
["light", "yes", "delayed", 0.4],
["light", "no", "on time", 0.7],
["light", "no", "delayed", 0.3],
["heavy", "yes", "on time", 0.4],
["heavy", "yes", "delayed", 0.6],
["heavy", "no", "on time", 0.5],
["heavy", "no", "delayed", 0.5],
], [rain.distribution, maintenance.distribution]), name="train")
```

BN in Python

```
# Track maintenance node is conditional on rain
maintenance = Node(ConditionalProbabilityTable([
["none", "yes", 0.4],
["none", "no", 0.6],
["light", "yes", 0.2],
["light", "no", 0.8],
["heavy", "yes", 0.1],
["heavy", "no", 0.9]
], [rain.distribution]), name="maintenance")
```

BN in Python

```
# Appointment node is conditional on train
appointment = Node(ConditionalProbabilityTable([
["on time", "attend", 0.9],
["on time", "miss", 0.1],
["delayed", "attend", 0.6],
["delayed", "miss", 0.4]
], [train.distribution]), name="appointment")
```

BN in Python

- Second, we create the model by adding all the nodes and then describing which node is the parent of which other node by adding edges between them

```
# Create a Bayesian Network and add states
model = BayesianNetwork()
model.add_states(rain, maintenance, train, appointment)
# Add edges connecting nodes
model.add_edge(rain, maintenance)
model.add_edge(rain, train)
model.add_edge(maintenance, train)
model.add_edge(train, appointment)
# Finalize model
model.bake()
```

BN in Python

- For asking how probable a certain event is, we run the model with the values we are interested in
  - In this example, we want to ask what is the probability that there is no rain, no track maintenance, the train is on time, and we attend the meeting

```
# Calculate probability for a given observation
probability = model.probability([["none", "no", "on time", "attend"]])
print(probability)
```

BN in Python

- We could also use the program to provide probability distributions for all variables given some observed evidence
  - In the following case, we know that the train was delayed
  - Given this information, we compute and print the probability distributions of the variables Rain, Maintenance, and Appointment

```
# Calculate predictions based on the evidence that the train was delayed
predictions = model.predict_proba({
"train": "delayed"
})
# Print predictions for each node
for node, prediction in zip(model.states, predictions):
    if isinstance(prediction, str):
        print(f"{node.name}: {prediction}")
    else:
        print(f"{node.name}")
for value, probability in prediction.parameters[0].items():
    print(f" {value}: {probability:.4f}")
```

Cons of Inference by Enumeration

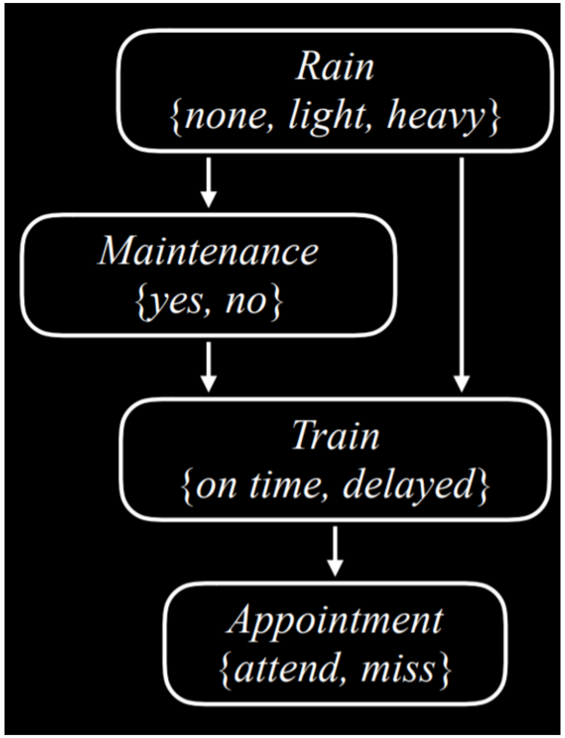
- This way of computing probability is inefficient
  - Think of many variables in the model
- A solution is to focus on an approximate inference instead of an exact inference
  - Some precision is lost, but often the imprecision is negligible
- Sampling is one technique of approximate inference
  - Each variable is sampled for a value according to its probability distribution

Sampling Example

- To generate a distribution using sampling with a die, we can roll the die multiple times and record what value we got each time
- Suppose we rolled the die 600 times
  - We count how many times we got 1, which is supposed to be roughly 100, and then repeat for the rest of the values, 2-6
  - Dividing each count by the total number of rolls will generate an approximate distribution of the values of rolling a die:
    - on one hand, it is unlikely that we get the result that each value has a probability of 1/6 of occurring (which is the exact probability), but we will get a value that's close to it

Approximate Inference

- Sampling
- Starting by sampling the Rain variable
  - the value none will be generated with a probability of 0.7
  - the value light will be generated with a probability of 0.2
  - the value heavy will be generated with a probability of 0.1



Approximate Inference

- Suppose that the sampled value we get is none
  - When we get to the Maintenance variable, we sample it, too, but only from the probability distribution where Rain is equal to none, because this is an already sampled result

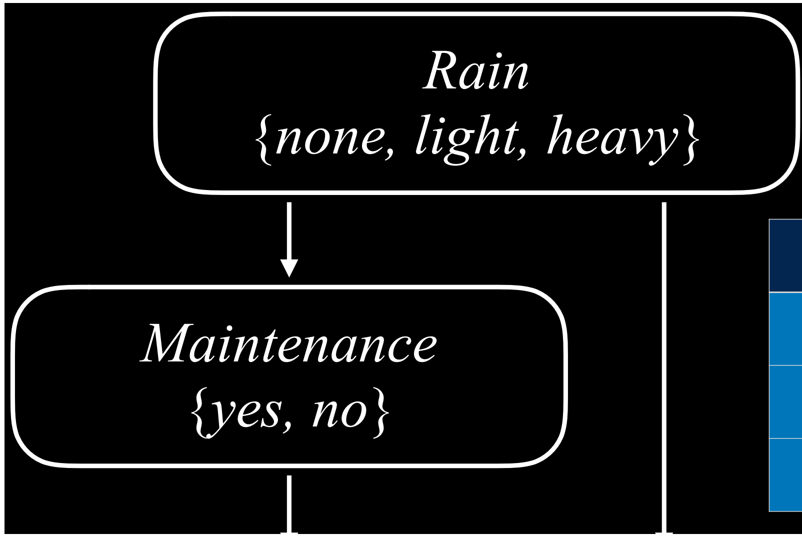
R = *none*

Rain
{*none*, *light*, *heavy*}

none	*light*	*heavy*
0.7	0.2	0.1

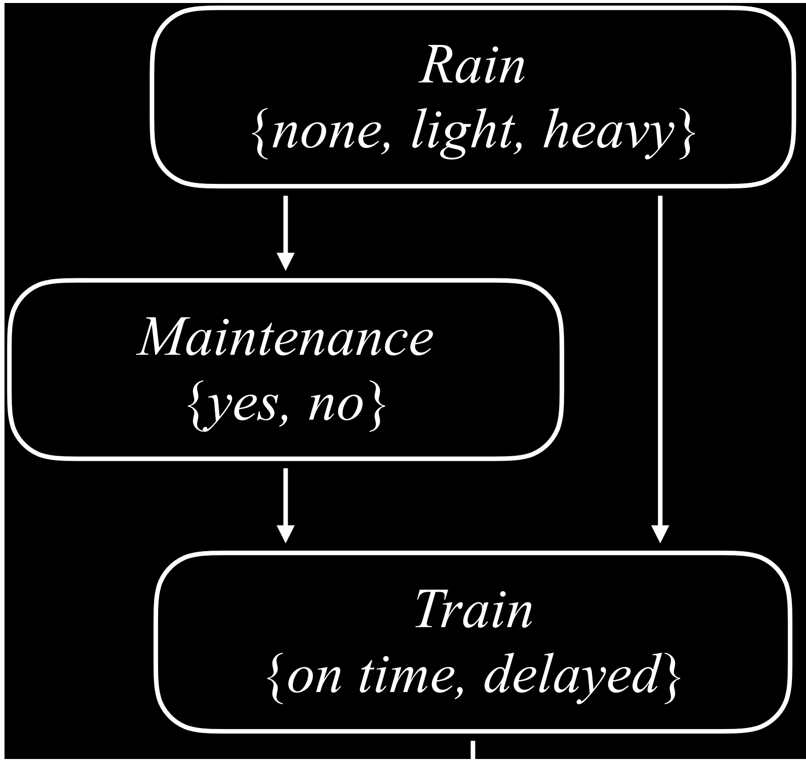
Approximate Inference

R = *none*
M = *yes*



R	*yes*	*no*
none	0.4	0.6
light	0.2	0.8
heavy	0.1	0.9

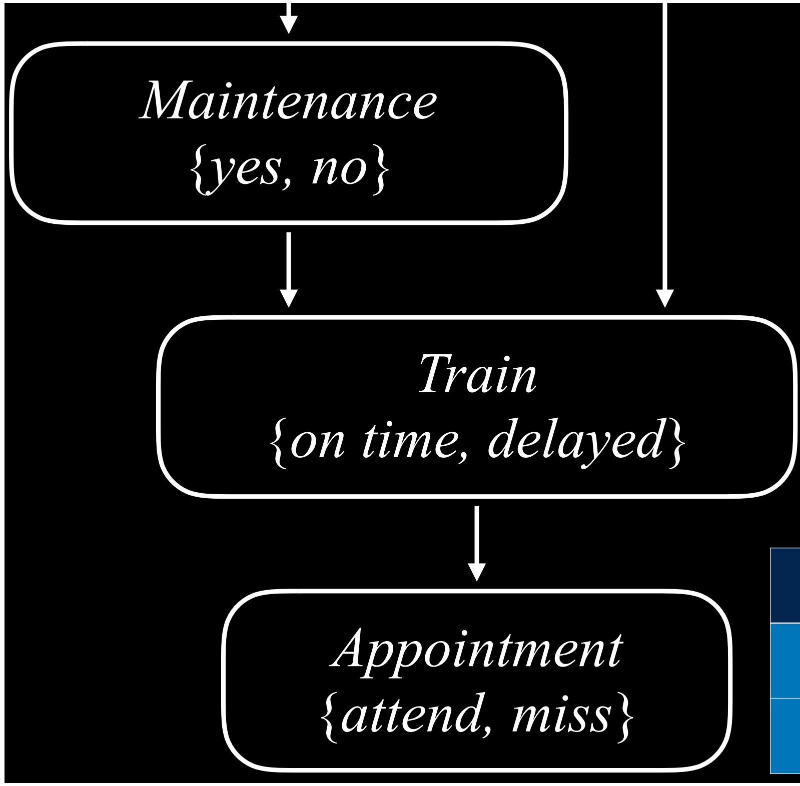
Approximate Inference



R = *none*
M = *yes*
T = *on time*

R	*M*	*on time*	*delayed*
none	*yes*	0.8	0.2
none	*no*	0.9	0.1
light	*yes*	0.6	0.4
light	*no*	0.7	0.3
heavy	*yes*	0.4	0.6
heavy	*no*	0.5	0.5

Approximate Inference



R = *none*
M = *yes*
T = *on time*
A = *attend*

T	*attend*	*miss*
on time	0.9	0.1
delayed	0.6	0.4

Approximate Inference

- Now we have one sample, and repeating this process multiple times generates a distribution

R = *light*	R = *light*	R = *none*	R = *none*
M = *no*	M = *yes*	M = *no*	M = *yes*
T = *on time*	T = *delayed*	T = *on time*	T = *on time*
A = *miss*	A = *attend*	A = *attend*	A = *attend*

R = *none*	R = *none*	R = *heavy*	R = *light*
M = *yes*	M = *yes*	M = *no*	M = *no*
T = *on time*	T = *on time*	T = *delayed*	T = *on time*
A = *attend*	A = *attend*	A = *miss*	A = *attend*

Approximate Inference

- P(Train = on time)?
  - Count the number of samples where Train has the value on time and divide by the number of samples

R = *light*	R = *light*	R = *none*	R = *none*
M = *no*	M = *yes*	M = *no*	M = *yes*
T = *on time*	T = *delayed*	T = *on time*	T = *on time*
A = *miss*	A = *attend*	A = *attend*	A = *attend*

R = *none*	R = *none*	R = *heavy*	R = *light*
M = *yes*	M = *yes*	M = *no*	M = *no*
T = *on time*	T = *on time*	T = *delayed*	T = *on time*
A = *attend*	A = *attend*	A = *miss*	A = *attend*

Approximate Inference

- We can also answer questions involving conditional probability, that is
  - P(Rain = light | Train = on time)?

R = *light*	R = *light*	R = *none*	R = *none*
M = *no*	M = *yes*	M = *no*	M = *yes*
T = *on time*	T = *delayed*	T = *on time*	T = *on time*
A = *miss*	A = *attend*	A = *attend*	A = *attend*
R = *none*	R = *none*	R = *heavy*	R = *light*
M = *yes*	M = *yes*	M = *no*	M = *no*
T = *on time*	T = *on time*	T = *delayed*	T = *on time*
A = *attend*	A = *attend*	A = *miss*	A = *attend*

Approximate Inference

- P(Rain = light | Train = on time)?
  - Ignore all samples where the value of Train is not on time (do not match the evidence) and proceed as before

R = *light*	R = *light*	R = *none*	R = *none*
M = *no*	M = *yes*	M = *no*	M = *yes*
T = *on time*	T = *delayed*	T = *on time*	T = *on time*
A = *miss*	A = *attend*	A = *attend*	A = *attend*

R = *none*	R = *none*	R = *heavy*	R = *light*
M = *yes*	M = *yes*	M = *no*	M = *no*
T = *on time*	T = *on time*	T = *delayed*	T = *on time*
A = *attend*	A = *attend*	A = *miss*	A = *attend*

Approximate Inference

- P(Rain = light | Train = on time)?
  - Count how many samples with Rain = light among those samples with Train = on time
    - then divide by the total number of samples where Train = on time

R = *light*	R = *light*	R = *none*	R = *none*
M = *no*	M = *yes*	M = *no*	M = *yes*
T = *on time*	T = *delayed*	T = *on time*	T = *on time*
A = *miss*	A = *attend*	A = *attend*	A = *attend*

R = *none*	R = *none*	R = *heavy*	R = *light*
M = *yes*	M = *yes*	M = *no*	M = *no*
T = *on time*	T = *on time*	T = *delayed*	T = *on time*
A = *attend*	A = *attend*	A = *miss*	A = *attend*

BN in Python: Sampling

```
import pomegranate
from collections import Counter
from model import model
def generate_sample():
# Mapping of random variable name to sample generated
sample = {}
# Mapping of distribution to sample generated
parents = {}
```

BN in Python: Sampling

```
# Loop over all states, assuming topological order

for state in model.states:
# If we have a non-root node, sample conditional on parents
    if isinstance(state.distribution, pomegranate.ConditionalProbabilityT:
        sample[state.name] = state.distribution.sample(parent_values=parents)
# Otherwise, just sample from the distribution alone
    else:
        sample[state.name] = state.distribution.sample()
# Keep track of the sampled value in the parents mapping
    parents[state.distribution] = sample[state.name]
# Return generated sample
return sample
```

BN in Python: Sampling

- To compute P(Appointment | Train = delayed), which is the probability distribution of the Appointment variable given that the train is delayed:

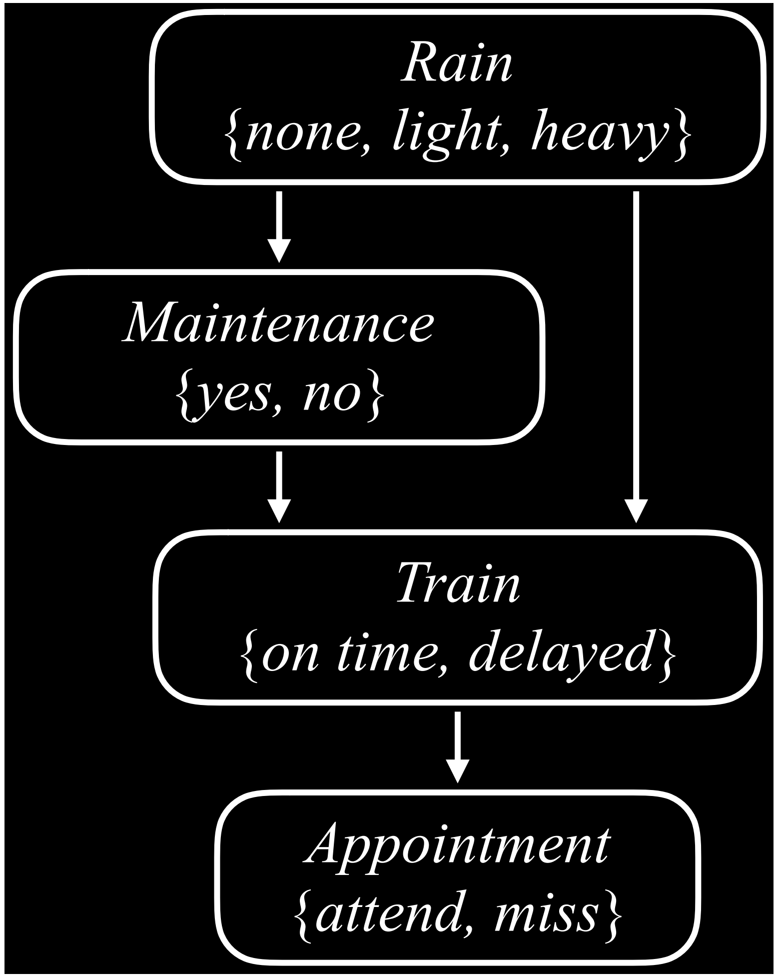
```
# Rejection sampling
# Compute distribution of Appointment given that train is delayed
N = 10000
data = []
# Repeat sampling 10,000 times
for i in range(N):
    # Generate a sample based on the function that we defined earlier
    sample = generate_sample()
    # If, in this sample, the variable of Train has the value delayed, save t
    if sample["train"] == "delayed":
        data.append(sample["appointment"])
    # Count how many times each value of the variable appeared. We can later norm
print(Counter(data))
```

Alternative Sampling

- Sampling by rejection rejects the samples that did not match the evidence
  - Inefficient!
- Likelihood Weighting
  - Start by fixing the values for evidence variables
  - Sample the non-evidence variables using conditional probabilities in the Bayesian Network
  - Weight each sample by its likelihood:
    - The probability of all the evidence

Example: Likelihood Weighting

- P(Rain = light | Train = on time)?
  - Start by fixing the evidence variable
    - Train = on time



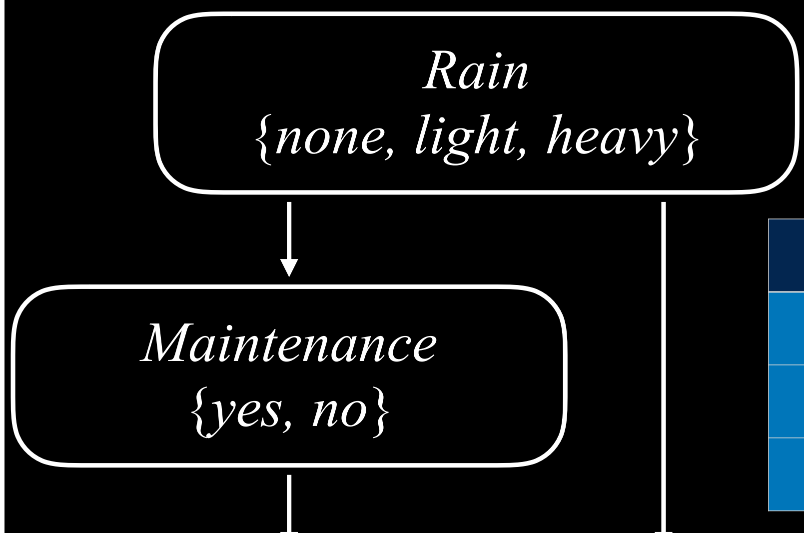
Example: Likelihood Weighting

R = *light*
T = *on time*

Rain
{*none*, *light*, *heavy*}

none	*light*	*heavy*
0.7	0.2	0.1

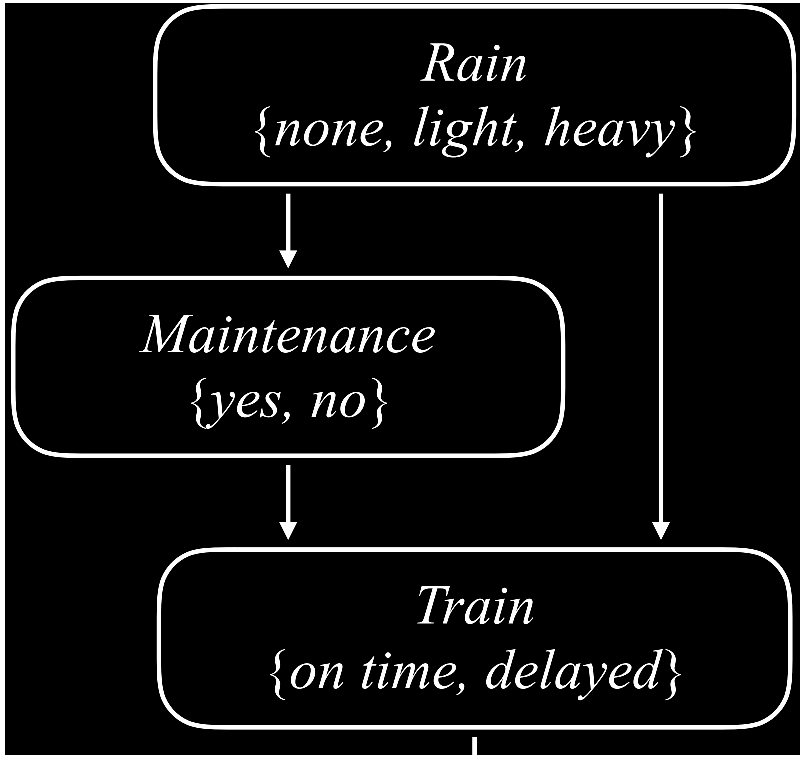
Example: Likelihood Weighting



R = *light*
M = *yes*
T = *on time*

R	*yes*	*no*
none	0.4	0.6
light	0.2	0.8
heavy	0.1	0.9

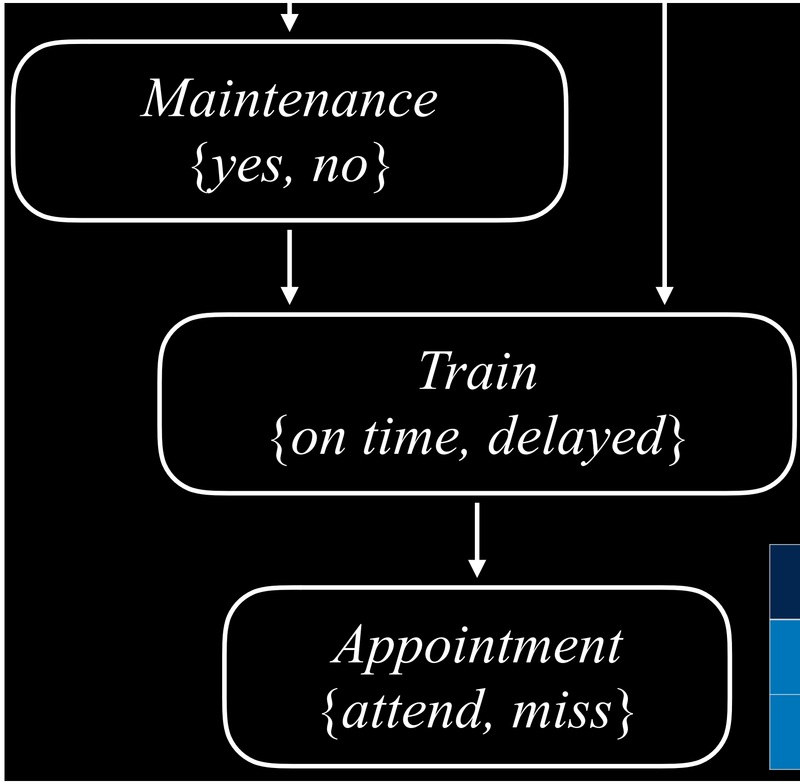
Example: Likelihood Weighting



R = *light*
M = *yes*
T = *on time*

R	*M*	*on time*	*delayed*
none	*yes*	0.8	0.2
none	*no*	0.9	0.1
light	*yes*	0.6	0.4
light	*no*	0.7	0.3
heavy	*yes*	0.4	0.6
heavy	*no*	0.5	0.5

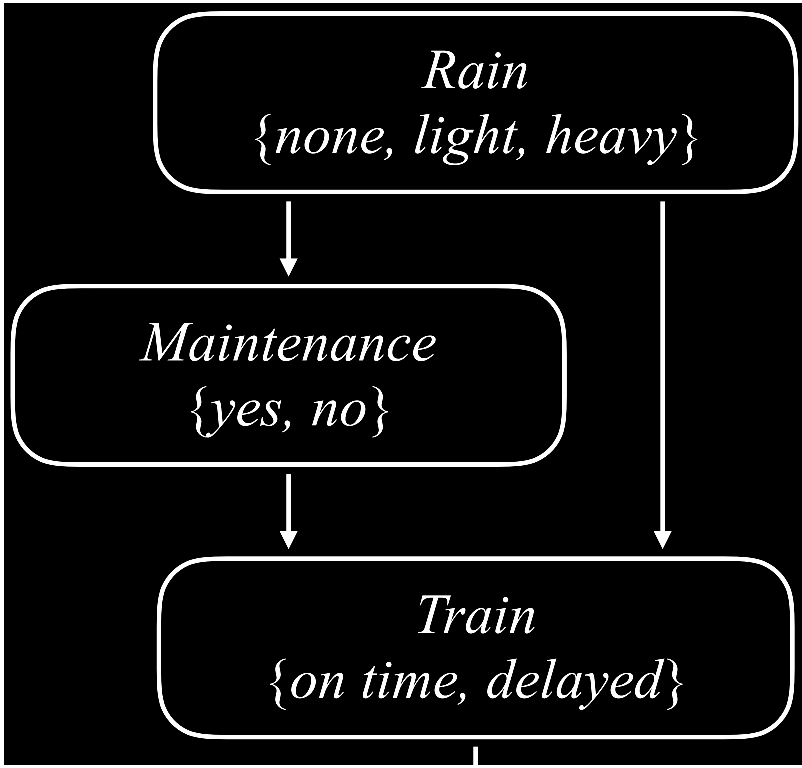
Example: Likelihood Weighting



R = *light*
M = *yes*
T = *on time*
A = *attend*

T	*attend*	*miss*
on time	0.9	0.1
delayed	0.6	0.4

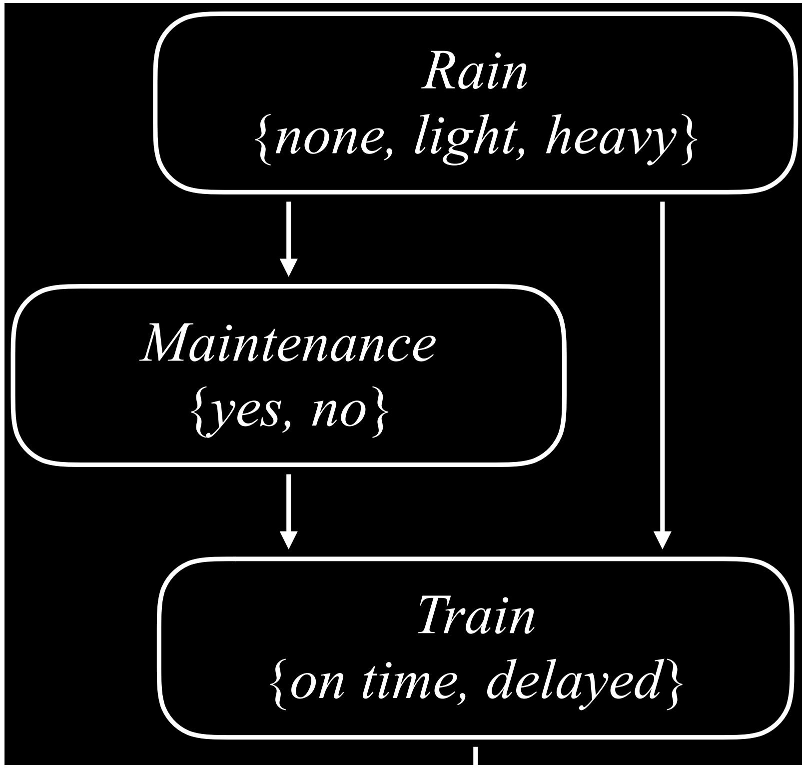
Example: Likelihood Weighting



R = *light*
M = *yes*
T = *on time*
A = *attend*

R	*M*	*on time*	*delayed*
none	*yes*	0.8	0.2
none	*no*	0.9	0.1
light	*yes*	0.6	0.4
light	*no*	0.7	0.3
heavy	*yes*	0.4	0.6
heavy	*no*	0.5	0.5

Example: Likelihood Weighting



R = *light*
M = *yes*
T = *on time*
A = *attend*

R	*M*	*on time*	*delayed*
none	*yes*	0.8	0.2
none	*no*	0.9	0.1
light	*yes*	0.6	0.4
light	*no*	0.7	0.3
heavy	*yes*	0.4	0.6
heavy	*no*	0.5	0.5

Uncertainty over Time

- Let's suppose to predict the weather at a given time based on the weather in the previous times (e.g., days)
- Xt: Weather at time t
- Markov assumption
  - The assumption that the current state depends on only a finite fixed number of previous states
- Markov chain
  - A sequence of random variables where the distribution of each variable follows the Markov assumption