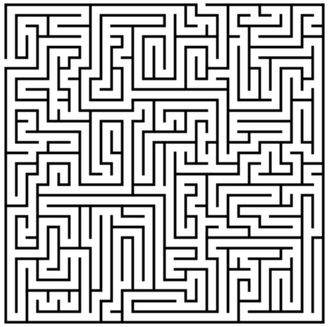
Uncertain knowledge and Reasoning

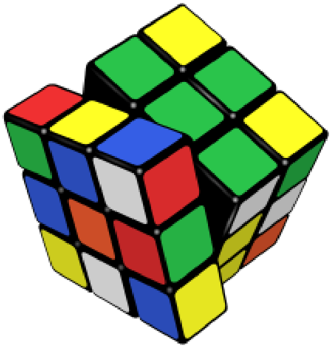
Artificial Intelligence

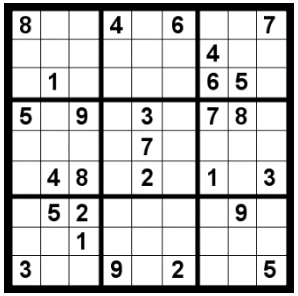
Slides are taken from Svetlana Lazebnik (UIUC)

Where are we?

- Now leaving: sequential, deterministic reasoning

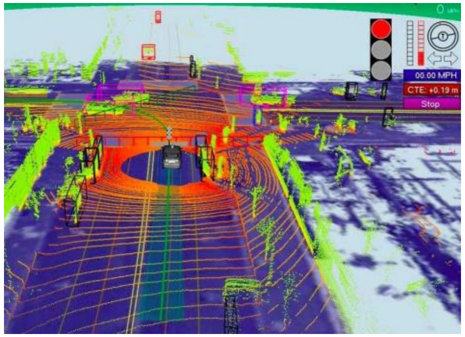


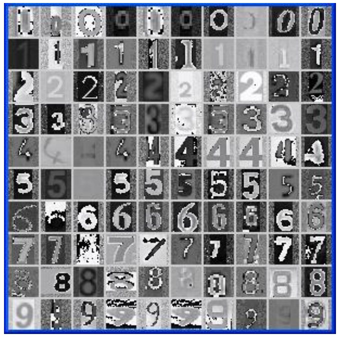




8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

- Entering: probabilistic reasoning and machine learning





Probability:
Review of main concepts

Making decisions under uncertainty

- Let action A_t = leave for airport t minutes before flight
  - Will A_t succeed, i.e., get me to the airport in time for the flight?
- Problems:
  - Partial observability (road state, other drivers' plans, etc.)
  - Noisy sensors (traffic reports)
  - Uncertainty in action outcomes (flat tire, etc.)
  - Complexity of modeling and predicting traffic
- Hence a non-probabilistic approach either
  - Risks falsehood: “A_{25} will get me there on time,” or
  - Leads to conclusions that are too weak for decision making:
    - A_{25} will get me there on time if there's no accident on the bridge and it doesn't rain and my tires remain intact, etc., etc.
    - A_{1440} will get me there on time but I’ll have to stay overnight in the airport

Making decisions under uncertainty

- Suppose the agent believes the following:

  $P(A_{25}$ gets me there on time$) = 0.04$

  $P(A_{90}$ gets me there on time$) = 0.70$

  $P(A_{120}$ gets me there on time$) = 0.95$

  $P(A_{1440}$ gets me there on time$) = 0.9999$

- Which action should the agent choose?
  - Depends on preferences for missing flight vs. time spent waiting
  - Encapsulated by a *utility function*
- The agent should choose the action that maximizes the *expected utility*:

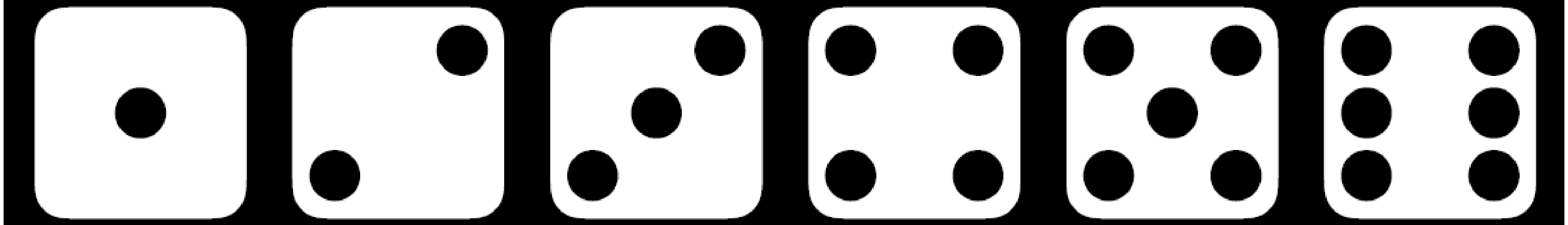
  $P(A_t$ succeeds$) * U(A_t$ succeeds$) + P(A_t$ fails$) * U(A_t$ fails$)$

Making decisions under uncertainty

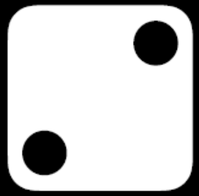
- More generally: the expected utility of an action is defined as:
  $EU(a) = \Sigma_{\text{outcomes of a}}\ P(\text{outcome}|a)\ U(\text{outcome})$

- **Utility theory** is used to represent and infer preferences
- **Decision theory** = probability theory + utility theory

- Possible Worlds



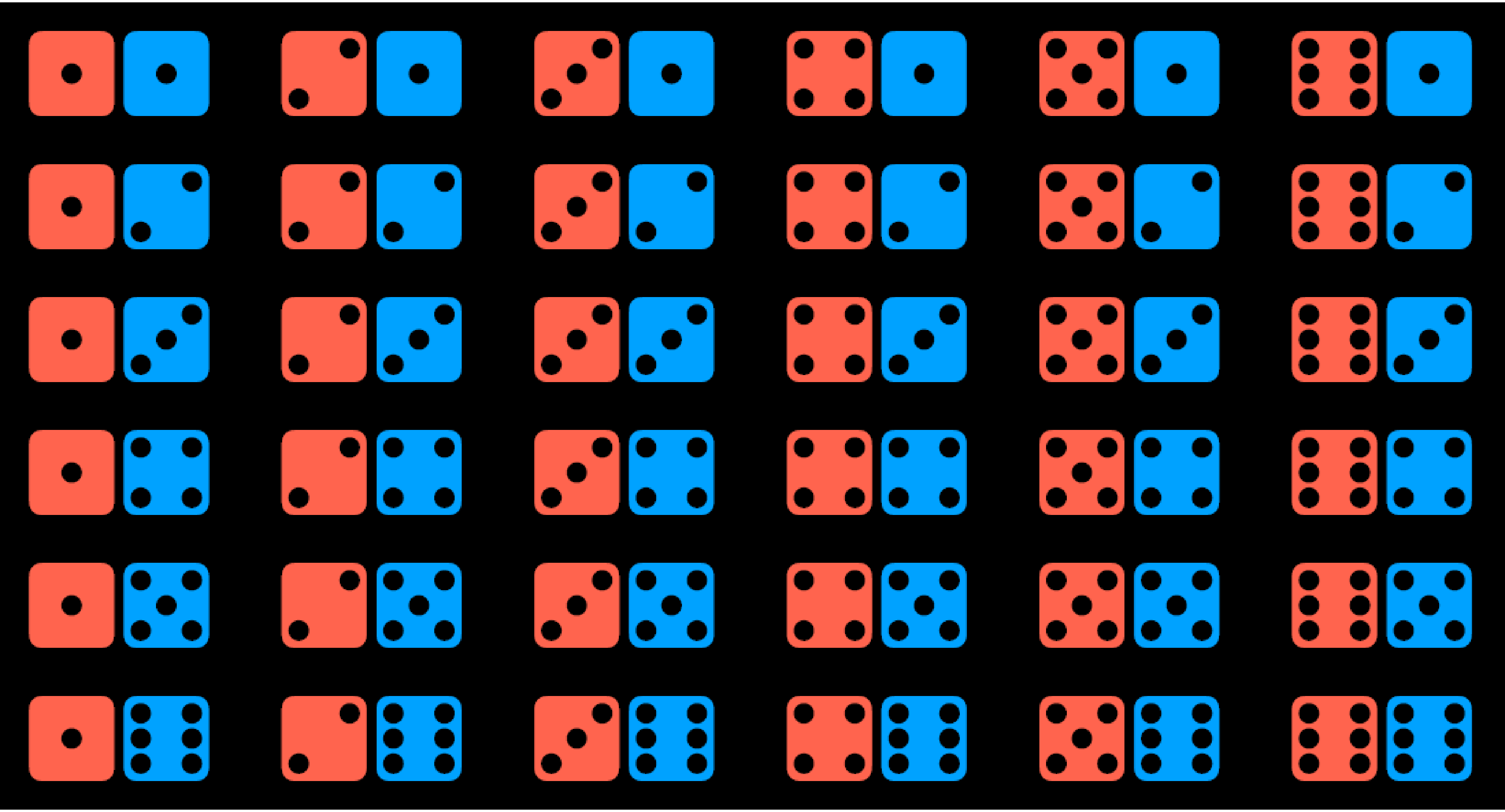
$\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$

$P($  $) = \frac{1}{6}$

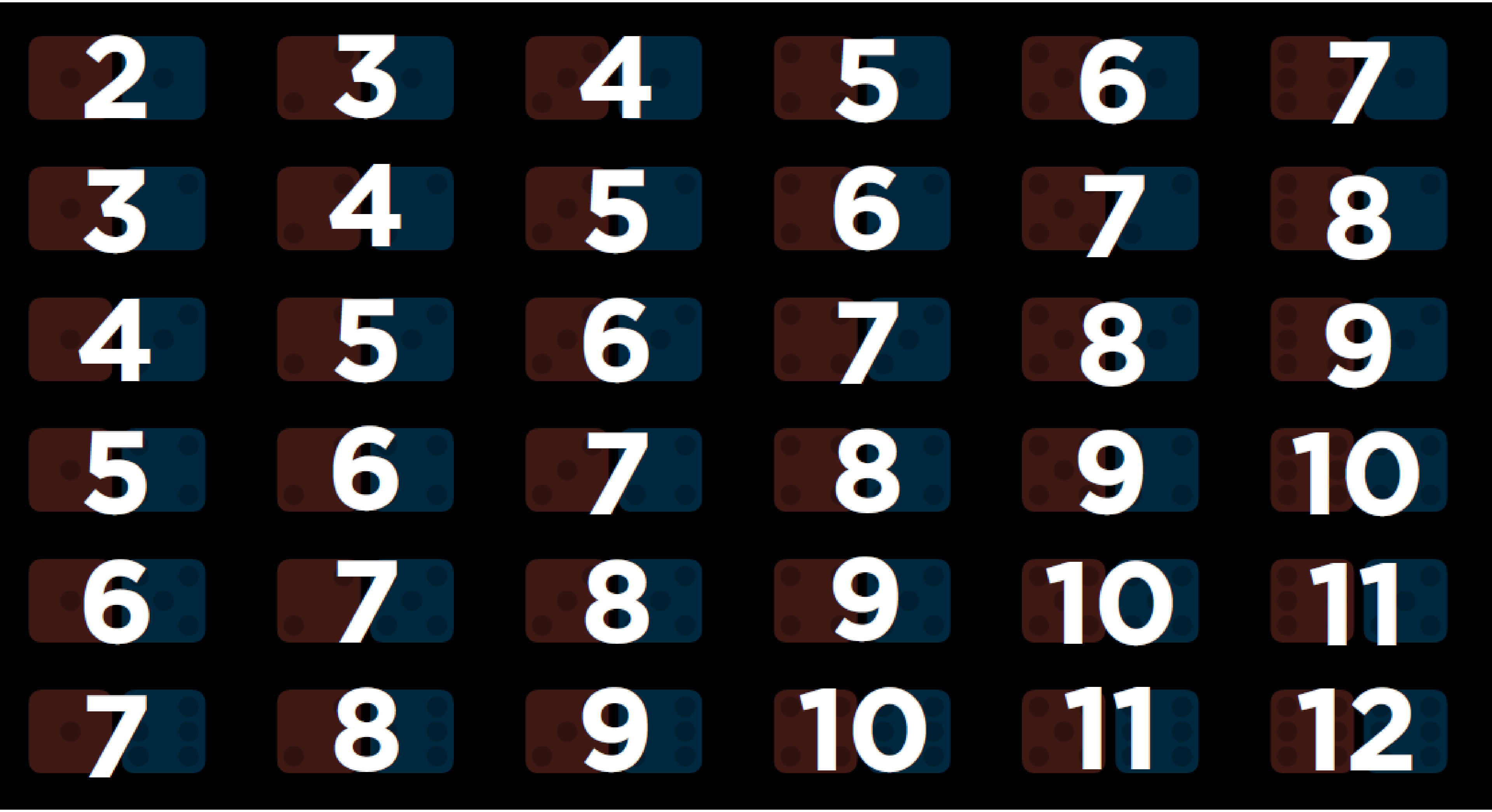
$$0 \leq P(\omega) \leq 1$$

$$\sum_{\omega \in \Omega} P(\omega) = 1$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

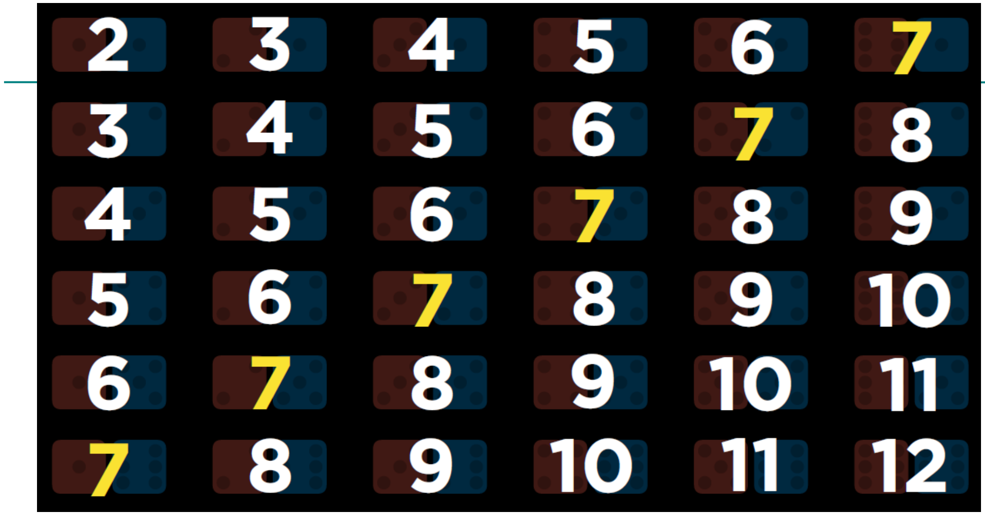


Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



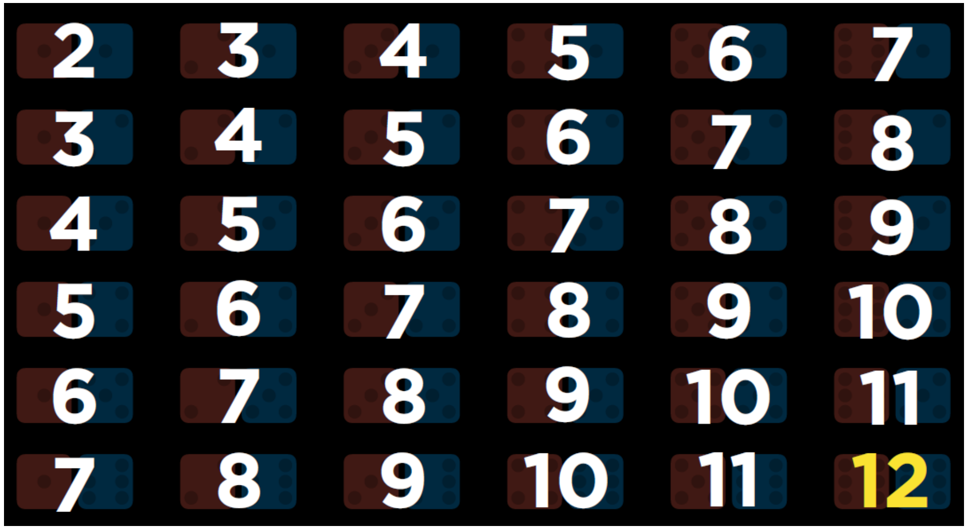
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10
6	7	8	9	10	11
7	8	9	10	11	12

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10
6	7	8	9	10	11
7	8	9	10	11	12

$$P(sum\ to\ 7) = \frac{6}{36} = \frac{1}{6}$$



2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10
6	7	8	9	10	11
7	8	9	10	11	12

$$P(sum\ to\ 12) = \frac{1}{36}$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Kolmogorov's axioms of probability

- For any propositions (events) a, b
  - $0 \leq P(a) \leq 1$
  - $P(\text{True}) = 1$ and $P(\text{False}) = 0$
  - $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$
    - Subtraction accounts for double-counting

- $P(\neg a) = 1 - P(a)$

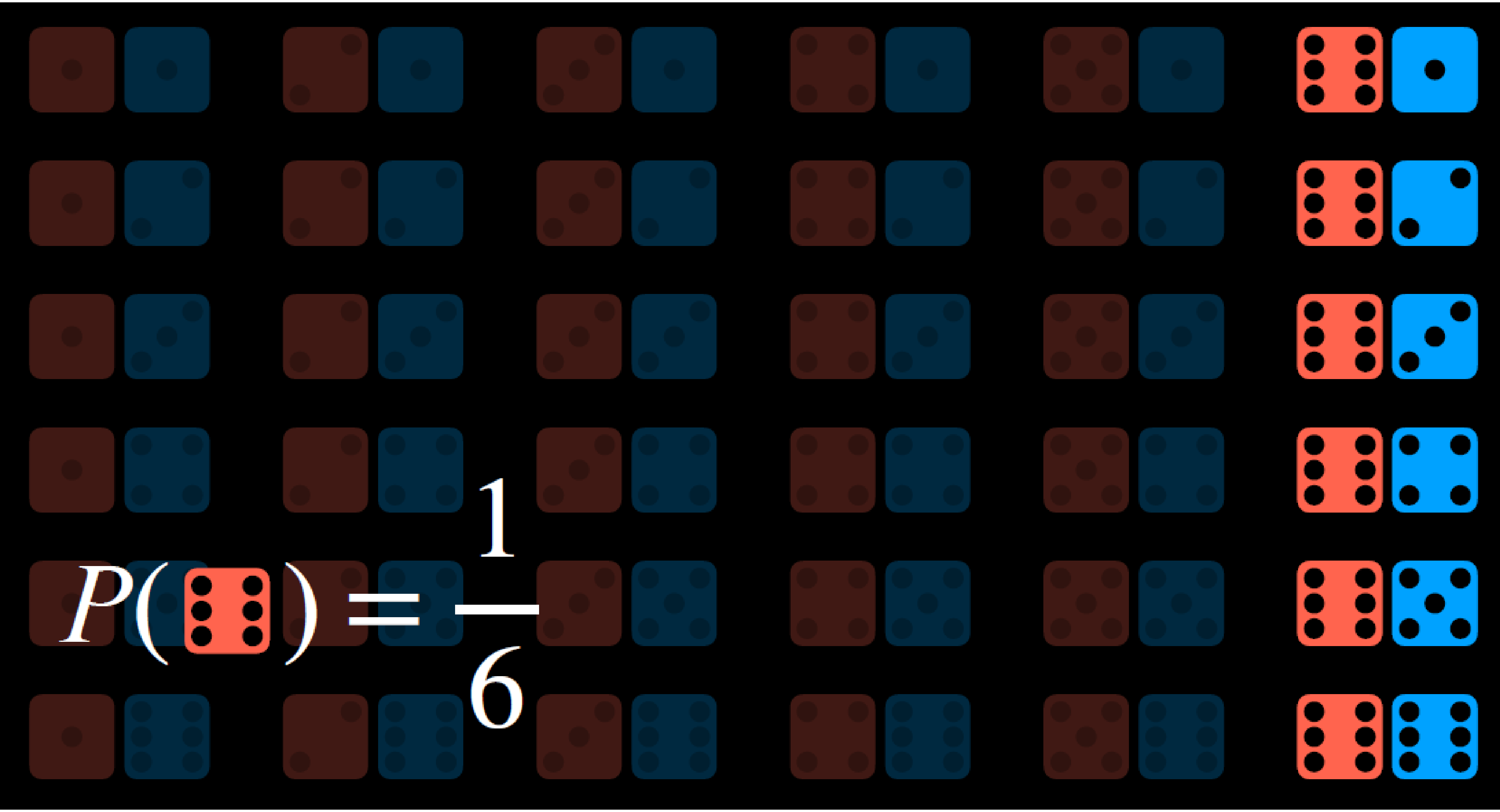
- unconditional probability degree of belief in a proposition in the absence of any other evidence

- conditional probability degree of belief in a proposition given some evidence that has already been revealed

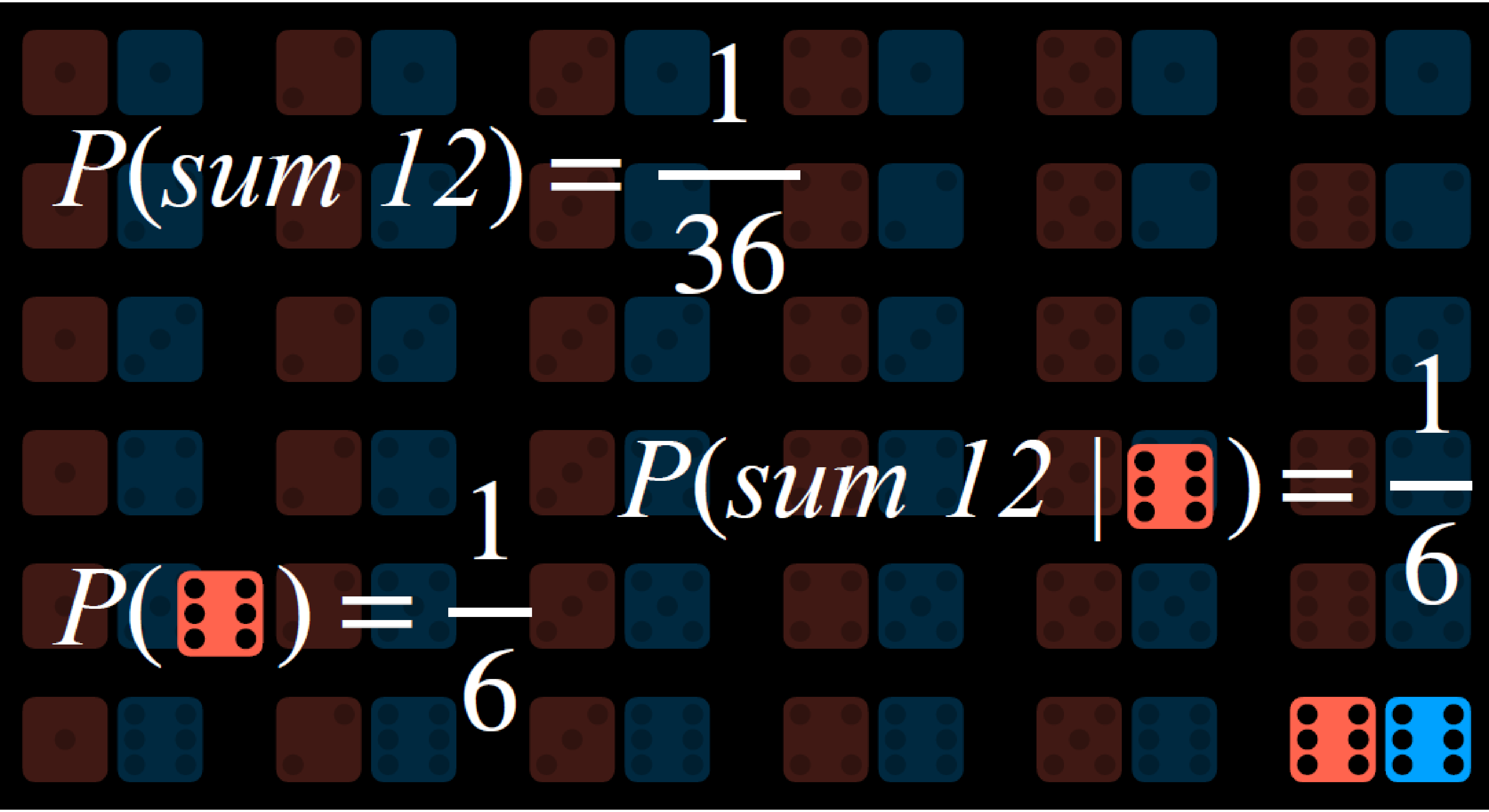
- $P(a \mid b)$

- *P*(*rain today* | *rain yesterday*)
- *P*(*route change* | *traffic conditions*)
- *P*(*disease* | *test results*)

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

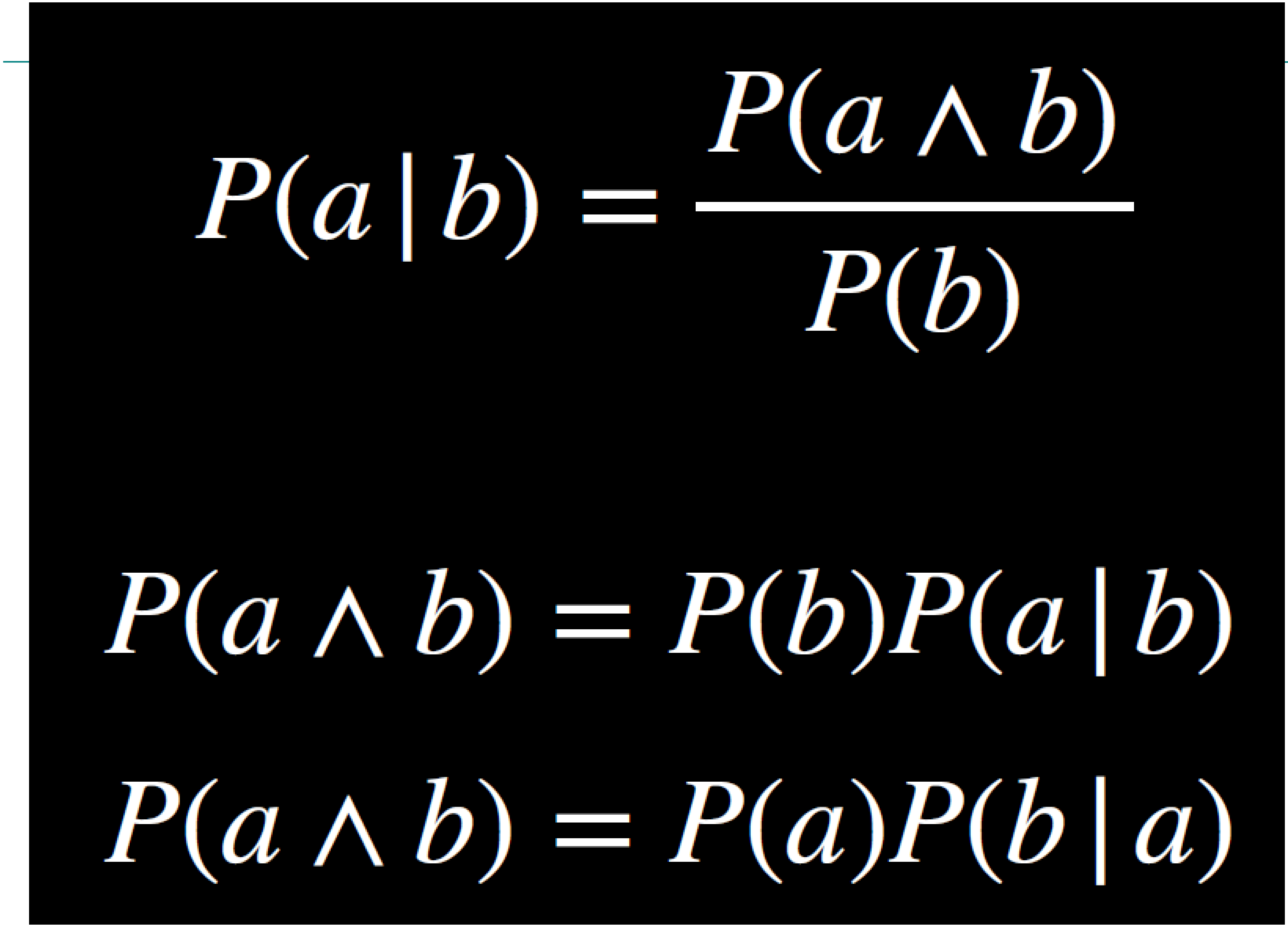


$$P(sum\ 12) = \frac{1}{36}$$

$$P(\text{⚅}) = \frac{1}{6}$$

$$P(sum\ 12 \mid \text{⚅}) = \frac{1}{6}$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



$$P(a \mid b) = \frac{P(a \wedge b)}{P(b)}$$

$$P(a \wedge b) = P(b)P(a \mid b)$$

$$P(a \wedge b) = P(a)P(b \mid a)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

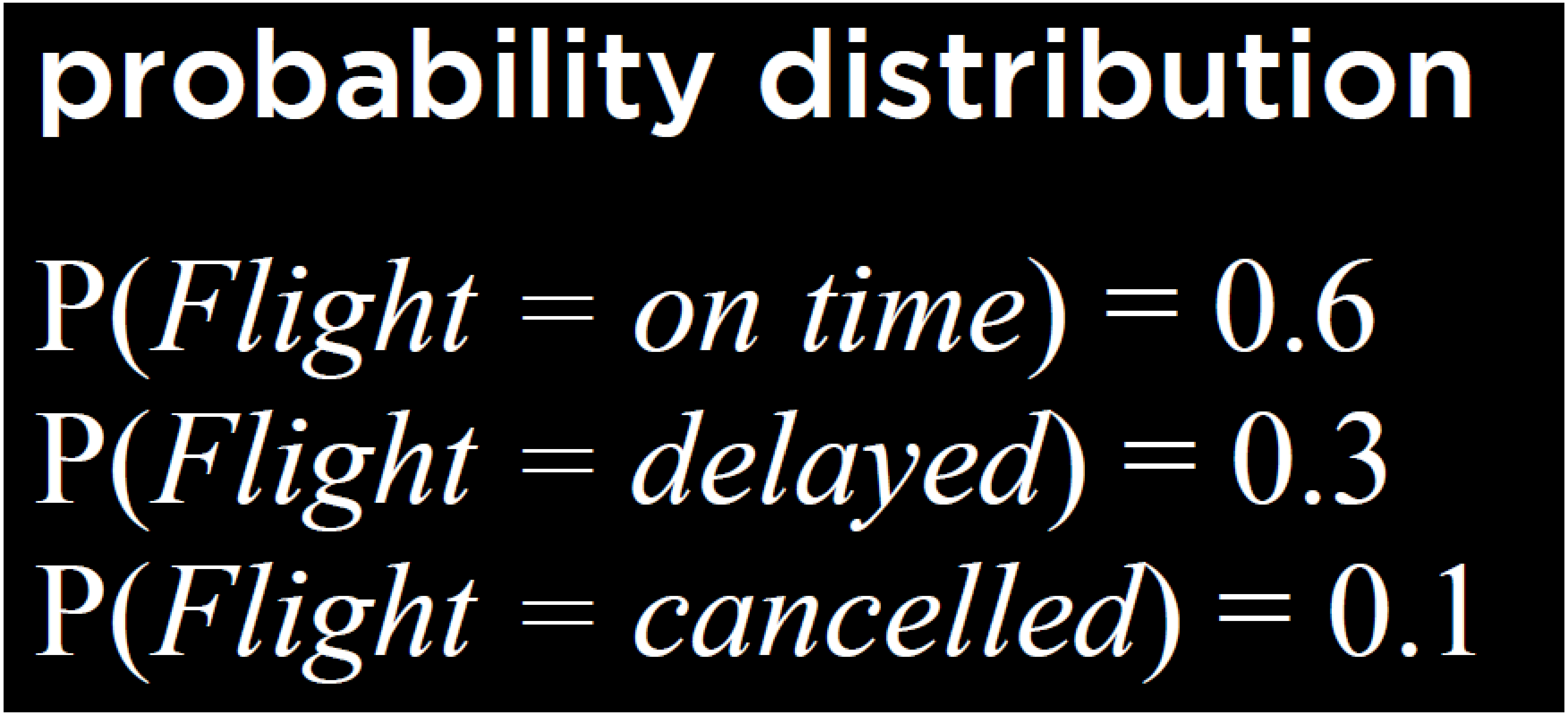
Random variables

- We describe the (uncertain) state of the world using ***random variables***
- **Random variable:** a variable in probability theory with a domain of possible values it can take on
  - Denoted by capital letters
  - – **R**: *Is it raining?*
  - – **W**: *What's the weather?*
  - – **D**: *What is the outcome of rolling two dice?*
  - – **S**: *What is the speed of my car (in MPH)?*
- Just like variables in CSPs, random variables take on values in a *domain*
  - Domain values must be *mutually exclusive* and *exhaustive*
  - – **R** in {True, False}
  - – **W** in {Sunny, Cloudy, Rainy, Snow}
  - – **D** in {(1,1), (1,2), … (6,6)}
  - – **S** in [0, 200]

Events

- Probabilistic statements are defined over *events*, or sets of world states
  - *"It is raining"*
  - *"The weather is either cloudy or snowy"*
  - *"The sum of the two dice rolls is 11"*
  - *"My car is going between 30 and 50 miles per hour"*
- Events are described using propositions about random variables:
  - $R = \text{True}$
  - $W = \text{"Cloudy"} \vee W = \text{"Snowy"}$
  - $D \in \{(5,6), (6,5)\}$
  - $30 \leq S \leq 50$
- Notation: $P(A)$ is the probability of the set of world states in which proposition A holds

Flight {on time, delayed, cancelled}

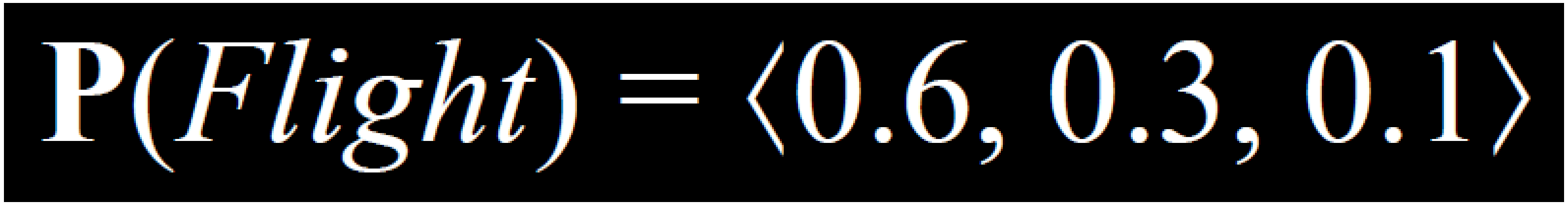


probability distribution

$$P(\mathit{Flight} = \mathit{on\ time}) = 0.6$$

$$P(\mathit{Flight} = \mathit{delayed}) = 0.3$$

$$P(\mathit{Flight} = \mathit{cancelled}) = 0.1$$



$$\mathbf{P}(\mathit{Flight}) = \langle 0.6, 0.3, 0.1 \rangle$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Atomic events

- ***Atomic event:*** a complete specification of the state of the world, or a complete assignment of domain values to all random variables
  - Atomic events are mutually exclusive and exhaustive

- E.g., if the world consists of only two Boolean variables *Cavity* and *Toothache*, then there are four distinct atomic events:

  $Cavity = false \wedge Toothache = false$
  $Cavity = false \wedge Toothache = true$
  $Cavity = true \wedge Toothache = false$
  $Cavity = true \wedge Toothache = true$

Joint probability distributions

- A ***joint distribution*** is an assignment of probabilities to every possible atomic event

Atomic event	**P**
Cavity = false ∧ *Toothache = false*	0.8
Cavity = false ∧ *Toothache = true*	0.1
Cavity = true ∧ *Toothache = false*	0.05
Cavity = true ∧ *Toothache = true*	0.05

  - Why does it follow from the axioms of probability that the probabilities of all possible atomic events must sum to 1?

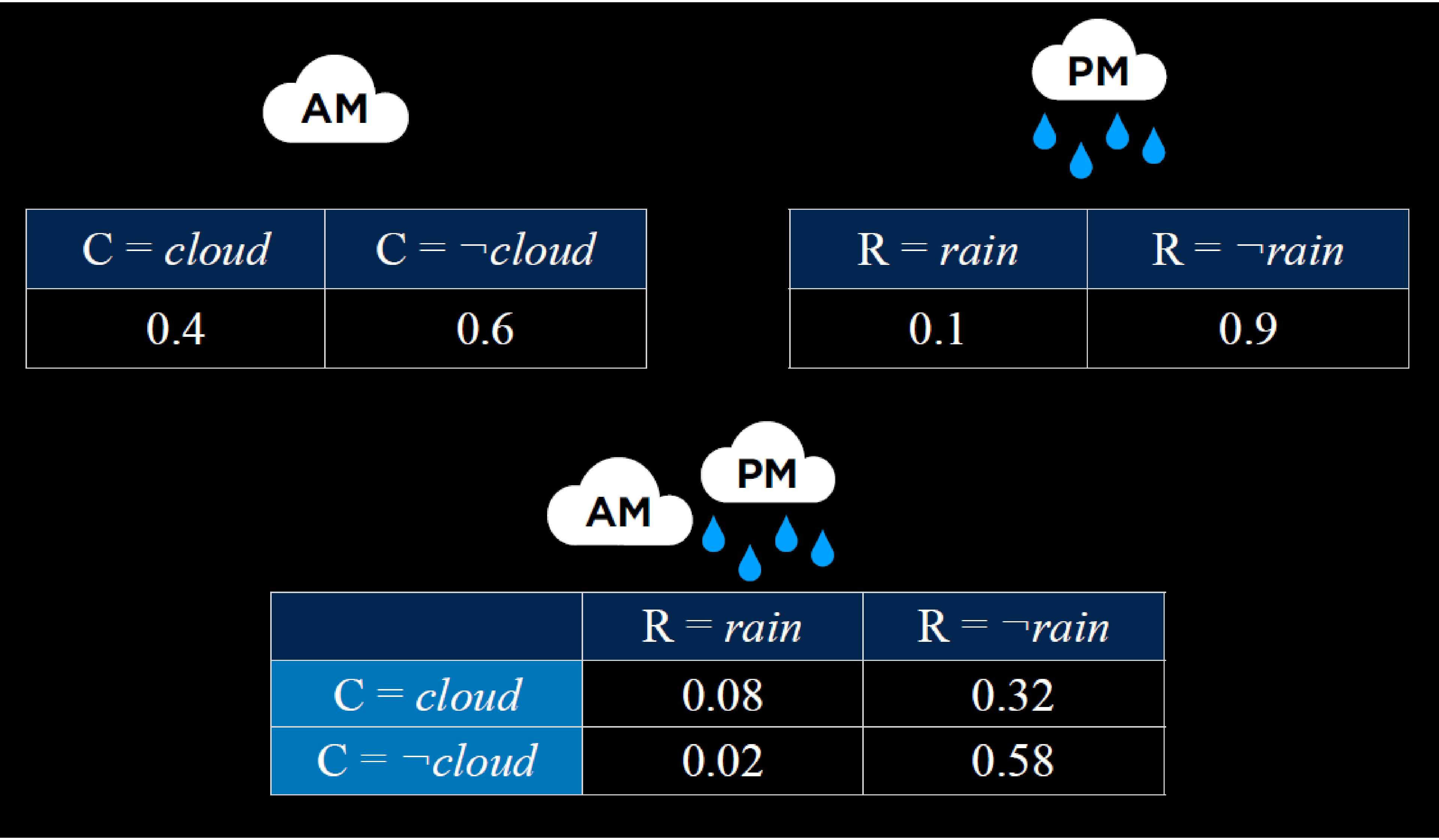
Joint probability distributions

- A ***joint distribution*** is an assignment of probabilities to every possible atomic event
- Suppose we have a joint distribution of n random variables with domain sizes d
  - What is the size of the probability table?
  - Impossible to write out completely for all but the smallest distributions

Notation

- $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$ refers to a single entry (atomic event) in the joint probability distribution table
  - Shorthand: $P(x_1, x_2, \ldots, x_n)$
- $P(X_1, X_2, \ldots, X_n)$ refers to the entire joint probability distribution table
- $P(x_1\ )$ can also refer to the probability of an event
  - E.g., $X_1 = x_1$ is an event

Joint Probability

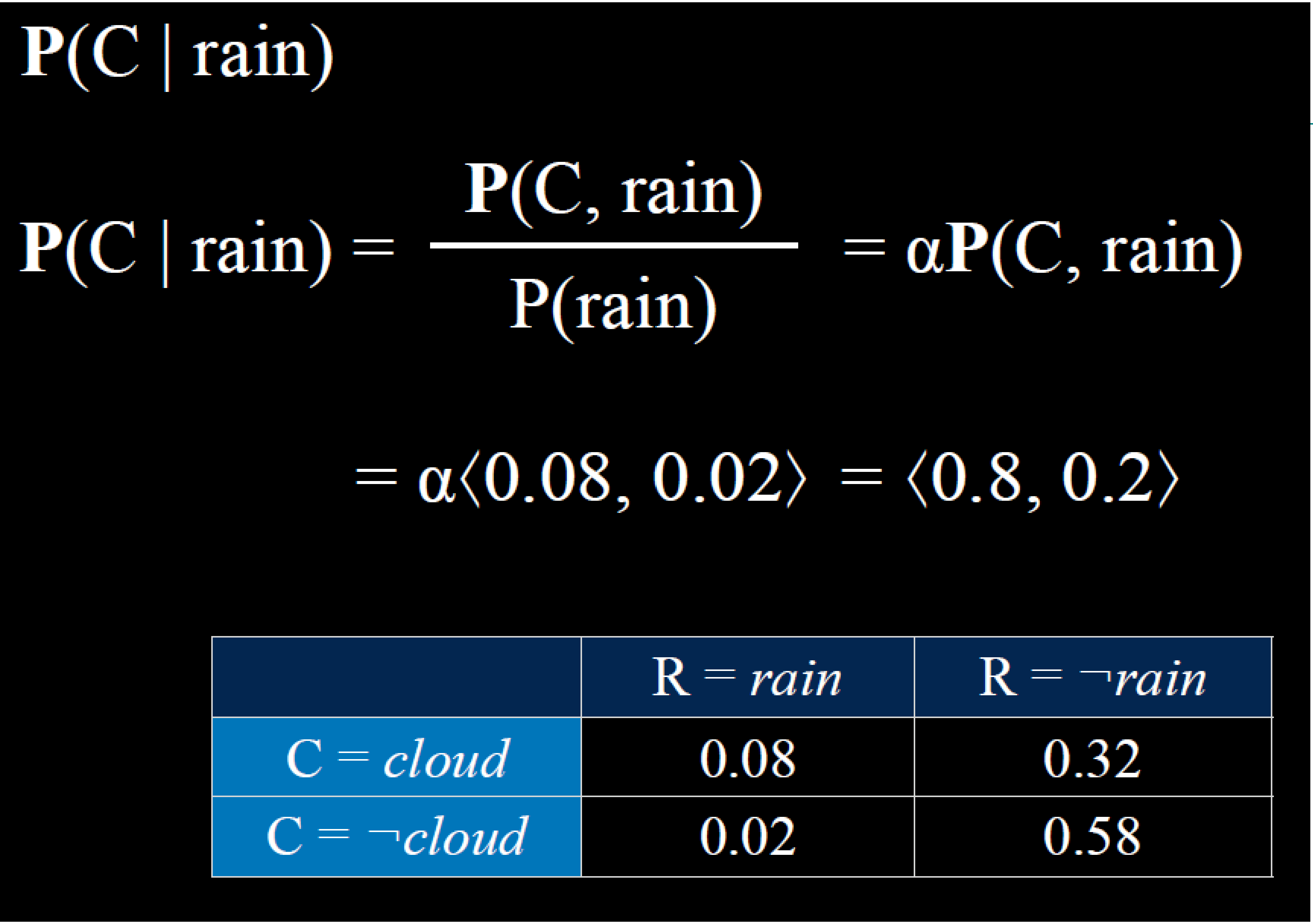


$C = cloud$	$C = \neg cloud$
0.4	0.6

$R = rain$	$R = \neg rain$
0.1	0.9

	$R = rain$	$R = \neg rain$
$C = cloud$	0.08	0.32
$C = \neg cloud$	0.02	0.58

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



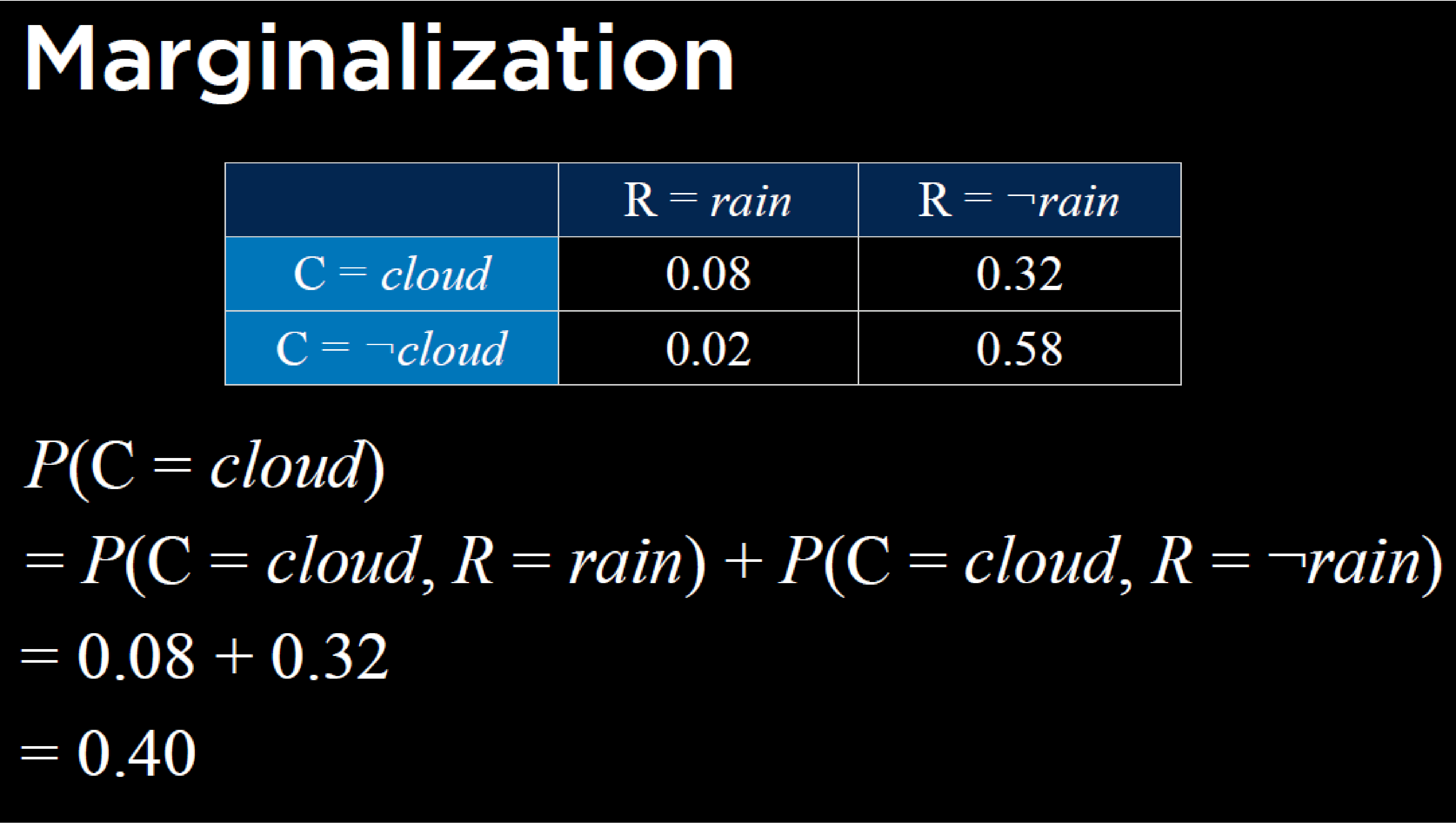
$\mathrm{P}(\mathrm{C} \mid \mathrm{rain})$

$$\mathrm{P}(\mathrm{C} \mid \mathrm{rain}) = \frac{\mathrm{P}(\mathrm{C}, \mathrm{rain})}{\mathrm{P}(\mathrm{rain})} = \alpha\mathrm{P}(\mathrm{C}, \mathrm{rain})$$

$$= \alpha\langle 0.08, 0.02\rangle = \langle 0.8, 0.2\rangle$$

	$\mathrm{R} = \mathit{rain}$	$\mathrm{R} = \neg\mathit{rain}$
$\mathrm{C} = \mathit{cloud}$	0.08	0.32
$\mathrm{C} = \neg\mathit{cloud}$	0.02	0.58

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Marginalization

	$R = rain$	$R = \neg rain$
$C = cloud$	0.08	0.32
$C = \neg cloud$	0.02	0.58

$$P(C = cloud)$$

$$= P(C = cloud, R = rain) + P(C = cloud, R = \neg rain)$$

$$= 0.08 + 0.32$$

$$= 0.40$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Marginalization

$$P(a) = P(a, b) + P(a, \neg b)$$

$$P(X = x_i) = \sum_j P(X = x_i, Y = y_j)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Conditioning

$$P(a) = P(a \mid b)P(b) + P(a \mid \neg b)P(\neg b)$$

$$P(X = x_i) = \sum_j P(X = x_i \mid Y = y_j)P(Y = y_j)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Marginal probability distributions

- From the joint distribution P(X,Y) we can find the ***marginal distributions*** P(X) and P(Y)

P(Cavity, Toothache)	
Cavity = false $\wedge$*Toothache = false*	0.8
Cavity = false $\wedge$ *Toothache = true*	0.1
Cavity = true $\wedge$ *Toothache = false*	0.05
Cavity = true $\wedge$ *Toothache = true*	0.05

P(Cavity)	
Cavity = false	?
Cavity = true	?

P(Toothache)	
Toothache = false	?
Toochache = true	?

Marginal probability distributions

- From the joint distribution P(X,Y) we can find the ***marginal distributions*** P(X) and P(Y)
- To find P(X = x), sum the probabilities of all atomic events where X = x:

$$P(X = x) = P\big((X = x \wedge Y = y_1) \vee \ldots \vee (X = x \wedge Y = y_n)\big)$$

$$= P\big((x, y_1) \vee \ldots \vee (x, y_n)\big) = \sum_{i=1}^{n} P(x, y_i)$$

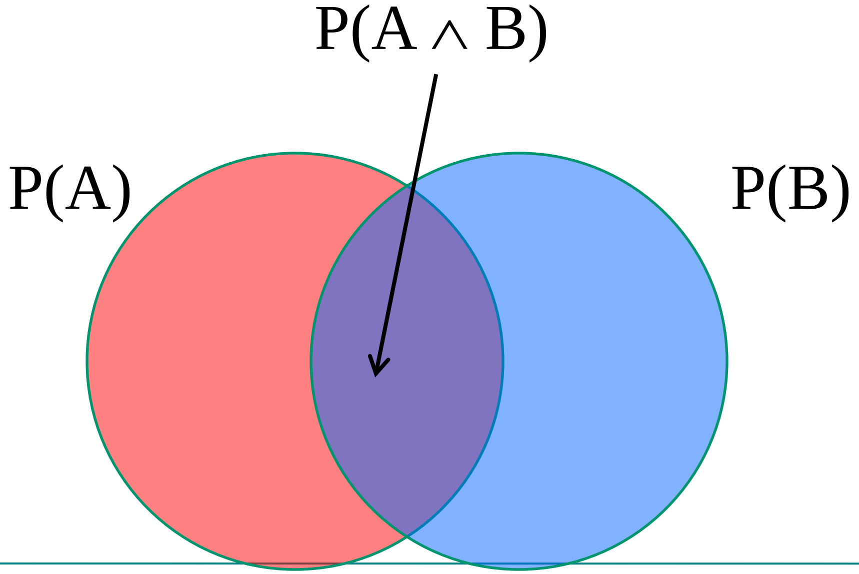
- This is called ***marginalization*** (we are *marginalizing out* all the variables except X)

Conditional probability

- Probability of cavity given toothache:
  P(*Cavity* = *true* | *Toothache* = *true*)

- For any two events A and B,

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(A, B)}{P(B)}$$



Conditional probability

P(Cavity, Toothache)	
$Cavity = false \land Toothache = false$	0.8
$Cavity = false \land Toothache = true$	0.1
$Cavity = true \land Toothache = false$	0.05
$Cavity = true \land Toothache = true$	0.05

P(Cavity)	
$Cavity = false$	0.9
$Cavity = true$	0.1

P(Toothache)	
$Toothache = false$	0.85
$Toothache = true$	0.15

- What is $P(Cavity = true \mid Toothache = false)$?

  0.05 / 0.85 = 0.059

- What is $P(Cavity = false \mid Toothache = true)$?

  0.1 / 0.15 = 0.667

Conditional distributions

- A conditional distribution is a distribution over the values of one variable given fixed values of other variables

P(Cavity, Toothache)	
Cavity = false ∧*Toothache = false*	0.8
Cavity = false ∧ *Toothache = true*	0.1
Cavity = true ∧ *Toothache = false*	0.05
Cavity = true ∧ *Toothache = true*	0.05

P(Cavity | Toothache = true)	
Cavity = false	0.667
Cavity = true	0.333

P(Cavity|Toothache = false)	
Cavity = false	0.941
Cavity = true	0.059

P(Toothache | Cavity = true)	
Toothache= false	0.5
Toothache = true	0.5

P(Toothache | Cavity = false)	
Toothache= false	0.889
Toothache = true	0.111

Normalization trick

- To get the whole conditional distribution $P(X \mid Y = y)$ at once, select all entries in the joint distribution table matching $Y = y$ and renormalize them to sum to one

P(Cavity, Toothache)	
Cavity = false ∧Toothache = false	0.8
Cavity = false ∧ Toothache = true	0.1
Cavity = true ∧ Toothache = false	0.05
Cavity = true ∧ Toothache = true	0.05

Select

Toothache, Cavity = false	
Toothache= false	0.8
Toothache = true	0.1

Renormalize

P(Toothache | Cavity = false)	
Toothache= false	0.889
Toothache = true	0.111

Normalization trick

- To get the whole conditional distribution P(X | Y = y) at once, select all entries in the joint distribution table matching Y = y and renormalize them to sum to one
- Why does it work?

$$\frac{P(x,y)}{\sum_{x'} P(x',y)} = \frac{P(x,y)}{P(y)}$$

Product rule

- Definition of conditional probability: $P(A\,|\,B) = \dfrac{P(A,B)}{P(B)}$

- Sometimes we have the conditional probability and want to obtain the joint:

$$P(A,B) = P(A\,|\,B)P(B) = P(B\,|\,A)P(A)$$

Chain rule

- Product rule:

$$P(A,B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

- Chain rule:

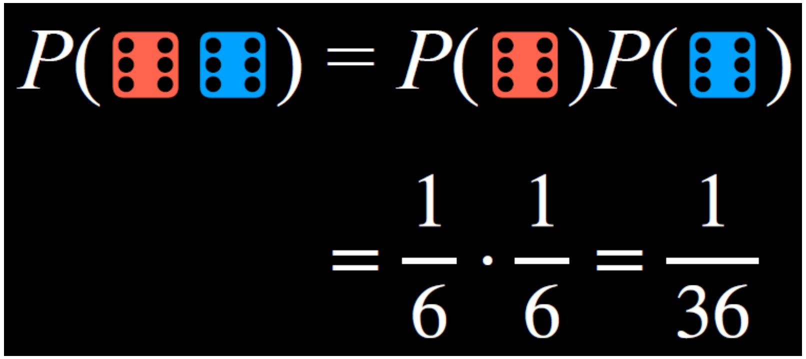
$$\begin{aligned} P(A_1,\ldots,A_n) &= P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1,A_2)\ldots P(A_n \mid A_1,\ldots,A_{n-1}) \\ &= \prod_{i=1}^{n} P(A_i \mid A_1,\ldots,A_{i-1}) \end{aligned}$$

Independence

- Two events A and B are *independent* if and only if $P(A \wedge B) = P(A, B) = P(A)\, P(B)$
  - In other words, $P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$
  - This is an important simplifying assumption for modeling, e.g., *Toothache* and *Weather* can be assumed to be independent
- Are two *mutually exclusive* events independent?
  - No, but for mutually exclusive events we have $P(A \vee B) = P(A) + P(B)$

Independence

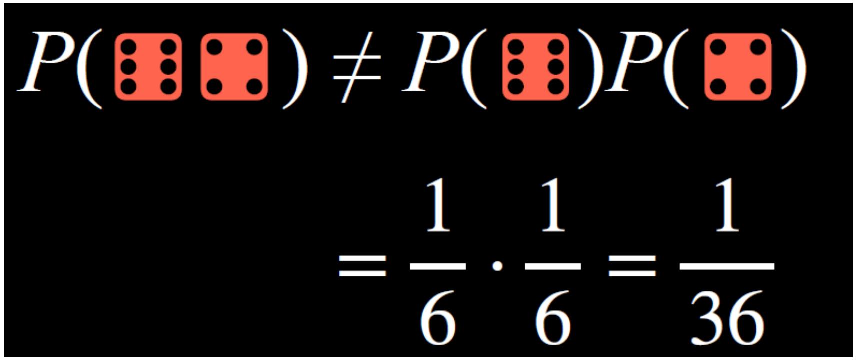
- the knowledge that one event occurs does not affect the probability of the other event



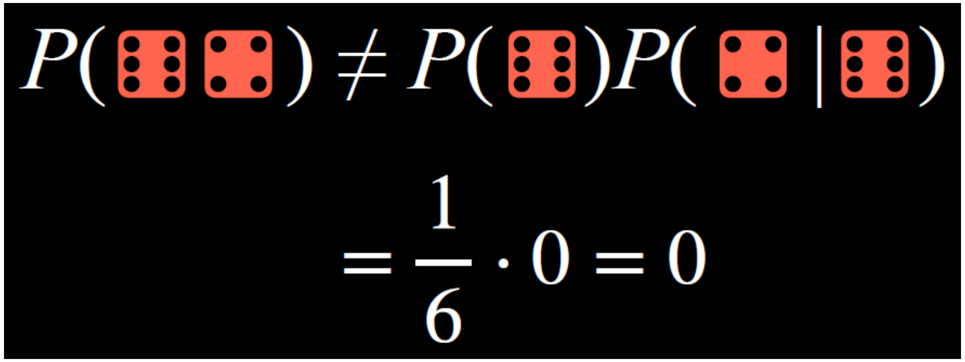
$$P(⚅⚅) = P(⚅)P(⚅) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$$

- $P(a \land b) = P(a)P(b|a)$

- $P(a \land b) = P(a)P(b)$



$$P(⚅⚃) \neq P(⚅)P(⚃) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$$



$$P(⚅⚃) \neq P(⚅)P(⚃|⚅) = \frac{1}{6} \cdot 0 = 0$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Independence

- Two events A and B are *independent* if and only if $P(A \wedge B) = P(A, B) = P(A)\, P(B)$
  - In other words, $P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$
  - This is an important simplifying assumption for modeling, e.g., *Toothache* and *Weather* can be assumed to be independent
- **Conditional independence**: A and B are *conditionally independent* given C iff $P(A \wedge B \mid C) = P(A \mid C)\, P(B \mid C)$
  - Equivalently: $P(A \mid B, C) = P(A \mid C)$ or $P(B \mid A, C) = P(B \mid C)$

Conditional independence: Example

- *Toothache*: boolean variable indicating whether the patient has a toothache
- *Cavity*: boolean variable indicating whether the patient has a cavity
- *Catch*: whether the dentist's probe catches in the cavity

- If the patient has a cavity, the probability that the probe catches in it doesn't depend on whether he/she has a toothache

  $P(Catch \mid Toothache, Cavity) = P(Catch \mid Cavity)$
- Therefore, *Catch* is conditionally independent of *Toothache* given *Cavity*
- Likewise, *Toothache* is conditionally independent of *Catch* given *Cavity*

  $P(Toothache \mid Catch, Cavity) = P(Toothache \mid Cavity)$
- Equivalent statement:

  $P(Toothache, Catch \mid Cavity) = P(Toothache \mid Cavity)\, P(Catch \mid Cavity)$

Conditional independence: Example

- How many numbers do we need to represent the joint probability table P(*Toothache, Cavity, Catch*)?

  $2^3 - 1 = 7$ independent entries

- Write out the joint distribution using chain rule:

  P(*Toothache, Catch, Cavity*)

  = P(*Cavity*) P(*Catch* | *Cavity*) P(*Toothache* | *Catch, Cavity*)

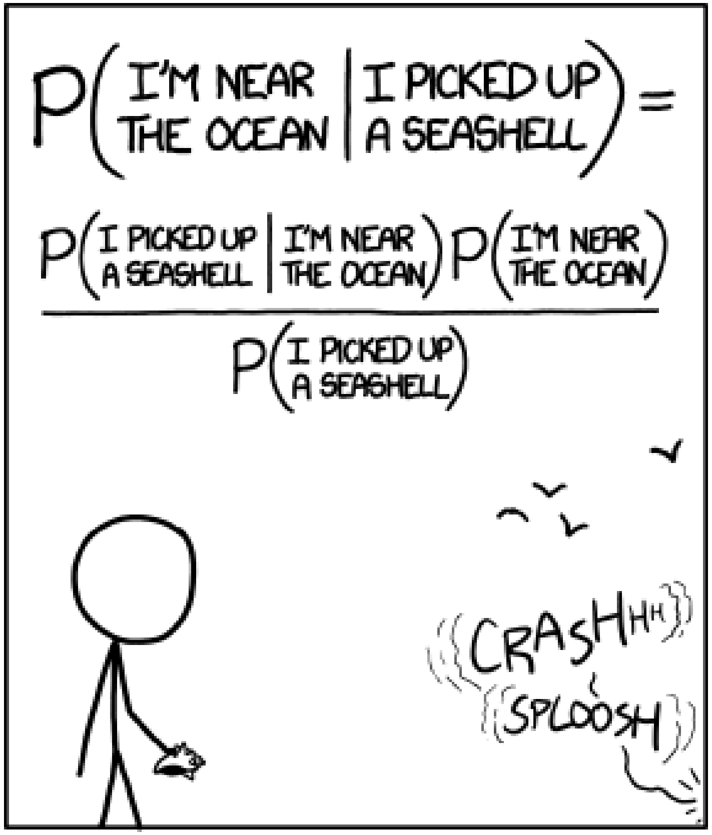
  = P(*Cavity*) P(*Catch* | *Cavity*) P(*Toothache* | *Cavity*)

- How many numbers do we need to represent these distributions?

  1 + 2 + 2 = 5 independent numbers

- In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in *n* to linear in *n*

Bayesian inference - Naïve Bayes model


P(I'M NEAR THE OCEAN | I PICKED UP A SEASHELL) =
P(I PICKED UP A SEASHELL | I'M NEAR THE OCEAN) P(I'M NEAR THE OCEAN)
P(I PICKED UP A SEASHELL)
CRASHHHH
SPLOOSH

STATISTICALLY SPEAKING, IF YOU PICK UP A SEASHELL AND *DON'T* HOLD IT TO YOUR EAR, YOU CAN PROBABLY HEAR THE OCEAN.

http://xkcd.com/1236/

Bayes' Rule

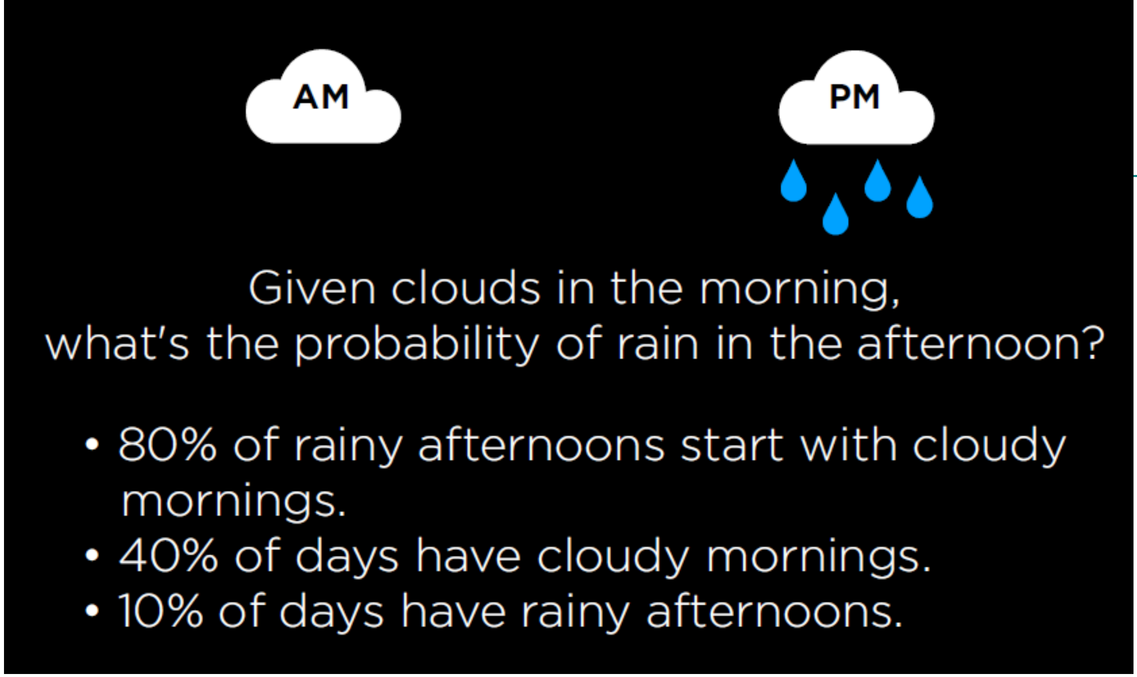
$$P(a \wedge b) = P(b)\, P(a \mid b)$$
$$P(a \wedge b) = P(a)\, P(b \mid a)$$

- The product rule gives us two ways to factor a joint probability:

$$P(A, B) = P(A \mid B) P(B) = P(B \mid A) P(A)$$

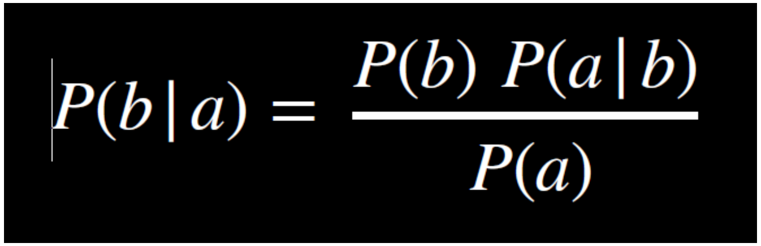
- Therefore, $$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

- Why is this useful?
  - Can update our beliefs about A based on evidence B
    - P(A) is the *prior* and P(A|B) is the *posterior*
  - Key tool for probabilistic inference: can get *diagnostic probability* from *causal probability*
    - E.g., P(Cavity = true | Toothache = true) from P(Toothache = true | Cavity = true)

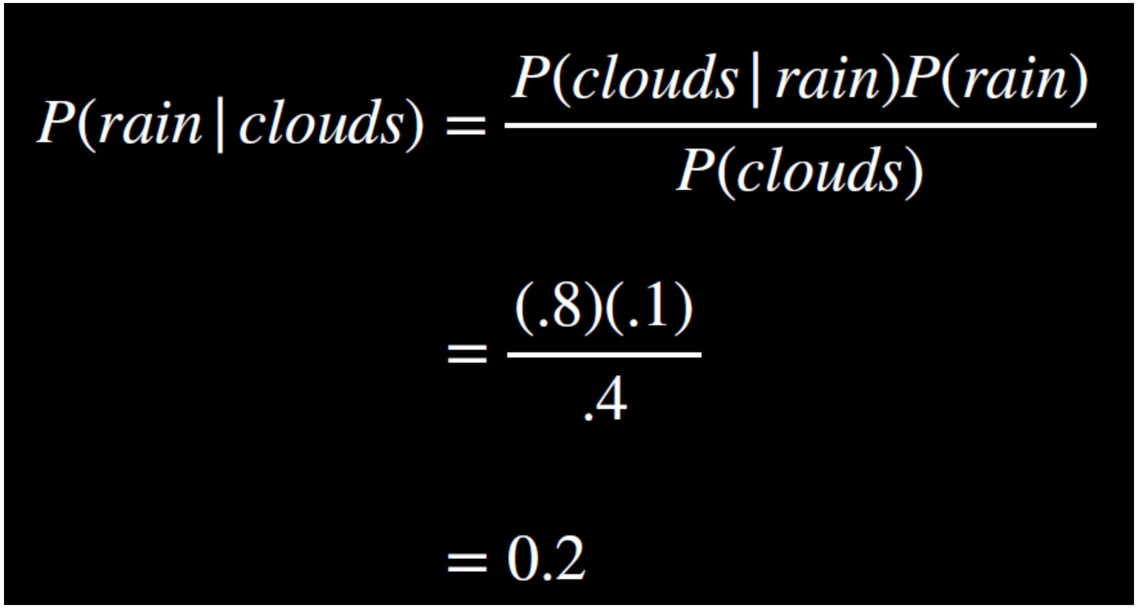


Given clouds in the morning,
what's the probability of rain in the afternoon?

- 80% of rainy afternoons start with cloudy mornings.
- 40% of days have cloudy mornings.
- 10% of days have rainy afternoons.



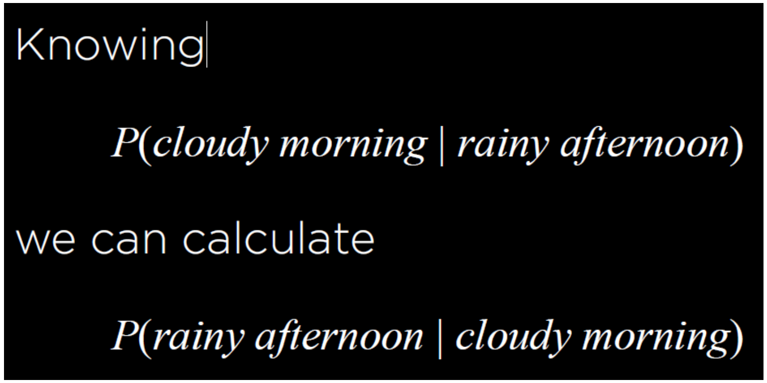
$$P(b \mid a) = \frac{P(b)\ P(a \mid b)}{P(a)}$$



$$P(rain \mid clouds) = \frac{P(clouds \mid rain) P(rain)}{P(clouds)}$$

$$= \frac{(.8)(.1)}{.4}$$

$$= 0.2$$



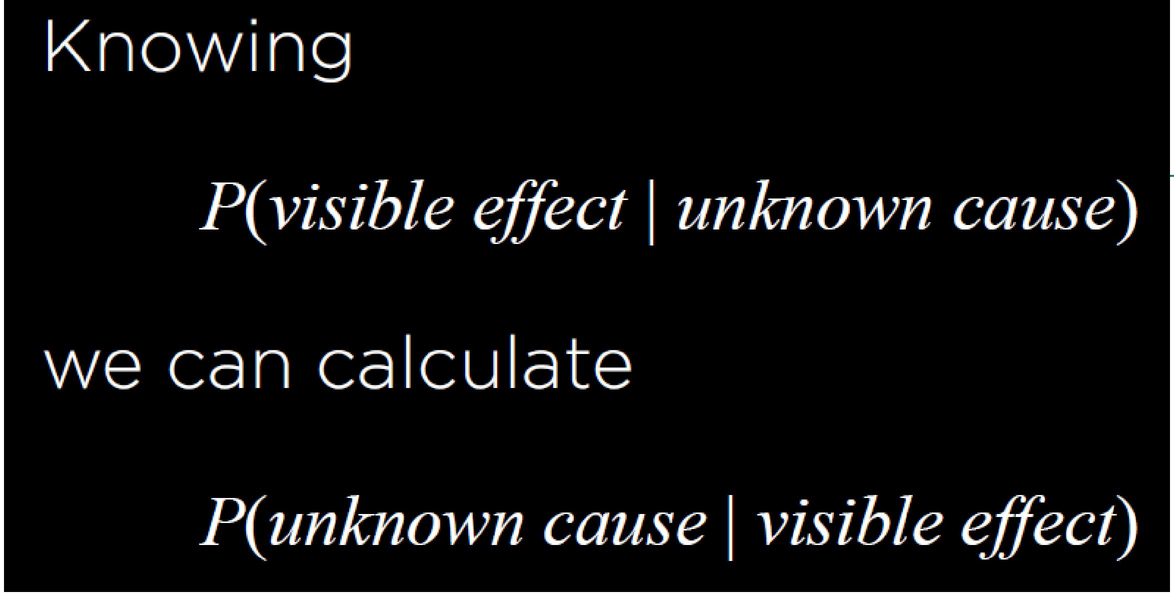
Knowing

$$P(cloudy\ morning \mid rainy\ afternoon)$$

we can calculate

$$P(rainy\ afternoon \mid cloudy\ morning)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

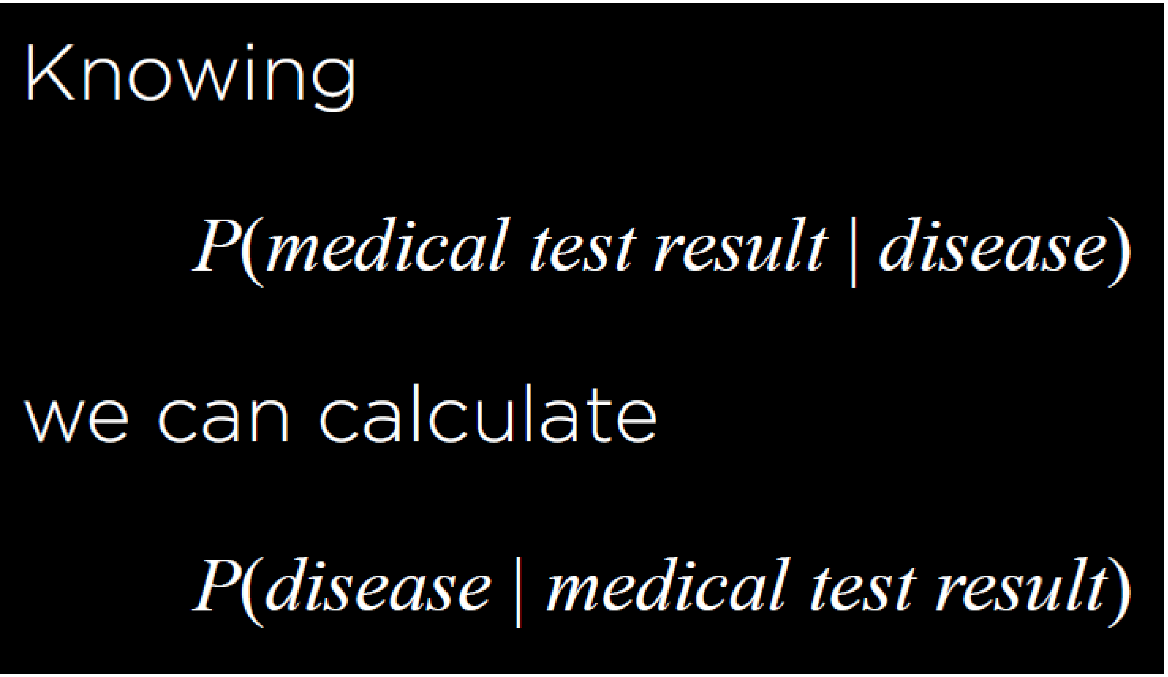


Knowing

$$P(\mathit{visible\ effect} \mid \mathit{unknown\ cause})$$

we can calculate

$$P(\mathit{unknown\ cause} \mid \mathit{visible\ effect})$$



Knowing

$$P(\mathit{medical\ test\ result} \mid \mathit{disease})$$

we can calculate

$$P(\mathit{disease} \mid \mathit{medical\ test\ result})$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Bayes Rule example

- Marie is getting married tomorrow, at an outdoor ceremony in the desert. In recent years, it has rained only 5 days each year (5/365 = 0.014). Unfortunately, the weatherman has predicted rain for tomorrow. When it actually rains, the weatherman correctly forecasts rain 90% of the time. When it doesn't rain, he incorrectly forecasts rain 10% of the time. What is the probability that it will rain on Marie's wedding?

$$P(\text{rain} \mid \text{predict}) = \frac{P(\text{predict} \mid \text{rain})P(\text{rain})}{P(\text{predict})}$$

$$= \frac{P(\text{predict} \mid \text{rain})P(\text{rain})}{P(\text{predict} \mid \text{rain})P(\text{rain}) + P(\text{predict} \mid \text{Ørain})P(\text{Ørain})}$$

Bayes Rule example

- Marie is getting married tomorrow, at an outdoor ceremony in the desert. In recent years, it has rained only 5 days each year (5/365 = 0.014). Unfortunately, the weatherman has predicted rain for tomorrow. When it actually rains, the weatherman correctly forecasts rain 90% of the time. When it doesn't rain, he incorrectly forecasts rain 10% of the time. What is the probability that it will rain on Marie's wedding?

$$P(\text{rain} \mid \text{predict}) = \frac{P(\text{predict} \mid \text{rain})P(\text{rain})}{P(\text{predict})}$$

$$= \frac{P(\text{predict} \mid \text{rain})P(\text{rain})}{P(\text{predict} \mid \text{rain})P(\text{rain}) + P(\text{predict} \mid \text{Ørain})P(\text{Ørain})}$$

$$= \frac{0.9 \times 0.014}{0.9 \times 0.014 + 0.1 \times 0.986} = \frac{0.0126}{0.0126 + 0.0986} = 0.111$$

Bayes rule: Example

- 1% of women at age forty who participate in routine screening have breast cancer. 80% of women with breast cancer will get positive mammographies. 9.6% of women without breast cancer will also get positive mammographies. A woman in this age group had a positive mammography in a routine screening. What is the probability that she actually has breast cancer?

$$P(\text{cancer} \mid \text{positive}) = \frac{P(\text{positive} \mid \text{cancer})P(\text{cancer})}{P(\text{positive})}$$

$$= \frac{P(\text{positive} \mid \text{cancer})P(\text{cancer})}{P(\text{positive} \mid \text{cancer})P(\text{cancer}) + P(\text{positive} \mid \text{Øcancer})P(\text{Øcancer})}$$

$$= \frac{0.8 \times 0.01}{0.8 \times 0.01 + 0.096 \times 0.99} = \frac{0.008}{0.008 + 0.095} = 0.0776$$

Law of total probability

$$P(X=x)=\sum_{i=1}^{n} P(X=x, Y=y_i)$$

$$=\sum_{i=1}^{n} P(X=x \mid Y=y_i)P(Y=y_i)$$

Probabilistic inference

- Suppose the agent has to make a decision about the value of an unobserved *query variable* X given some observed *evidence variable(s)* E = e
  - Partially observable, stochastic, episodic environment
  - Examples: X = {spam, not spam}, e = email message
    X = {zebra, giraffe, hippo}, e = image features

✗

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

✗

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY $99

✓

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

© Reuters

MAP decision

- Value x of X that has the highest posterior probability given the evidence E = e:

$$\hat{x} = \arg\max_x P(X = x \mid E = e) = \frac{P(E = e \mid X = x)P(X = x)}{P(E = e)}$$

$$\propto \arg\max_x P(E = e \mid X = x)P(X = x)$$

$$\underbrace{P(x \mid e)}_{\text{posterior}} \propto \underbrace{P(e \mid x)}_{\text{likelihood}}\underbrace{P(x)}_{\text{prior}}$$

- Maximum likelihood (ML) decision:

$$\hat{x} = \arg\max_x P(e \mid x)$$

Naïve Bayes model

- Suppose we have many different types of observations (symptoms, features) $E_1, \ldots, E_n$ that we want to use to obtain evidence about an underlying hypothesis X
- MAP decision:

$$P(X = x \mid E_1 = e_1, \ \ldots, E_n = e_n)$$
$$\mu \ P(X = x) P(E_1 = e_1, \ \ldots, E_n = e_n \mid X = x)$$

- We can make the simplifying assumption that the different features are conditionally independent *given the hypothesis*:

$$P(E_1 = e_1, \ \ldots, E_n = e_n \mid X = x) = \tilde{O}_{i=1}^{n} P(E_i = e_i \mid X = x)$$

  – If each feature can take on d values, what is the complexity of storing the resulting distributions?

Naïve Bayes model

- Posterior:

$$
\begin{aligned}
&P(X = x \mid E_1 = e_1, \; \ldots, E_n = e_n) \\
&\qquad \mu \; P(X = x) P(E_1 = e_1, \; \ldots, E_n = e_n \mid X = x) \\
&\qquad = P(X = x) \overset{n}{\underset{i=1}{\tilde{O}}} P(E_i = e_i \mid X = x)
\end{aligned}
$$

- MAP decision:

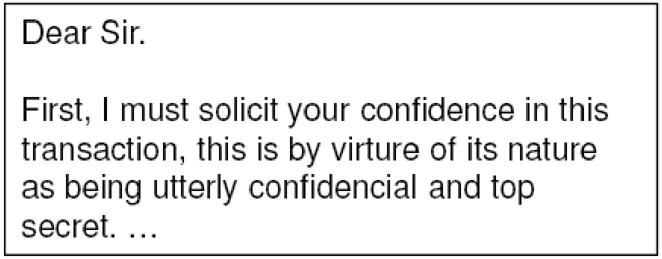
$$\hat{x} = \mathrm{argmax}_x \underbrace{P(x \mid e)}_{\text{posterior}} \; \mu \; \underbrace{P(x)}_{\text{prior}} \underbrace{\overset{n}{\underset{i=1}{\tilde{O}}} P(e_i \mid x)}_{\text{likelihood}}$$

Case study: Text document classification

- **MAP decision:** assign a document to the class with the highest posterio P(class | document)

- Example: spam classification
  - Classify a message as spam if P(spam | message) > P(¬spam | message)

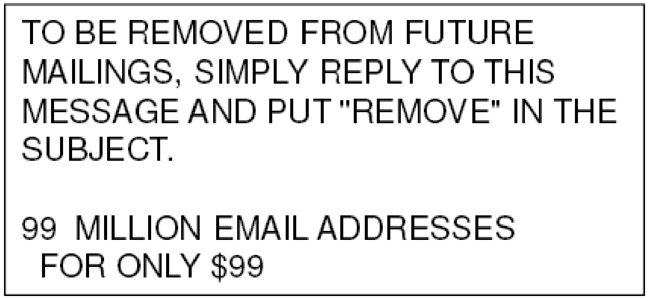
✗



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

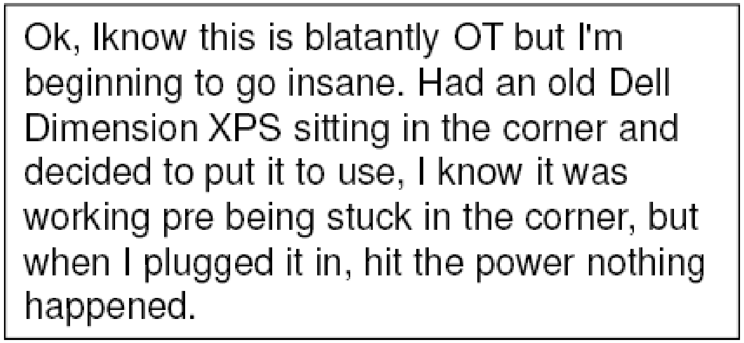
✗



TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY $99

✓



Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

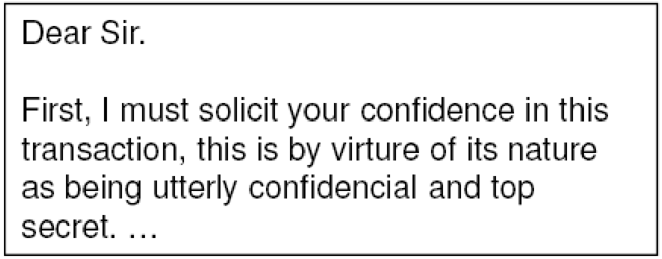
Case study: Text document classification

- **MAP decision:** assign a document to the class with the highest posterior $P(\text{class} \mid \text{document})$
- We have $P(\text{class} \mid \text{document}) \propto P(\text{document} \mid \text{class})P(\text{class})$
- To enable classification, we need to be able to estimate the likelihoods $P(\text{document} \mid \text{class})$ for all classes and
  priors $P(\text{class})$

Naïve Bayes Representation

- Goal: estimate likelihoods P(document | class) and priors P(class)
- Likelihood: ***bag of words*** representation
  - The document is a sequence of words $(w_1, \ldots, w_n)$
  - The order of the words in the document is not important
  - Each word is conditionally independent of the others given document class

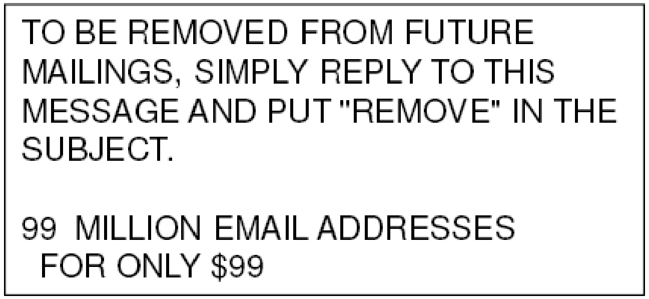
✗



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

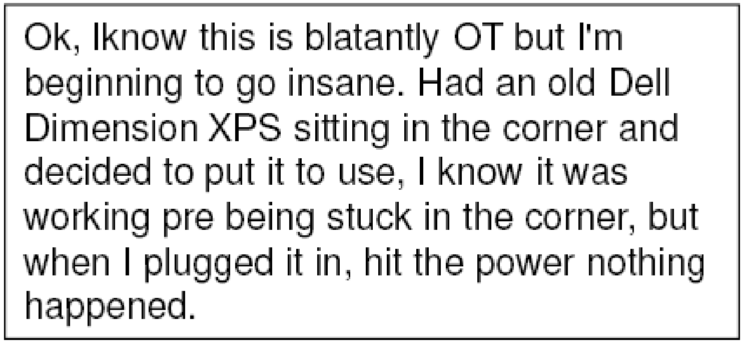
✗



TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY $99

✓



Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Naïve Bayes Representation

- Goal: estimate likelihoods P(document | class) and priors P(class)
- Likelihood: ***bag of words*** representation
  - The document is a sequence of words $(w_1, \ldots, w_n)$
  - The order of the words in the document is not important
  - Each word is conditionally independent of the others given document class

$$P(document \mid class) = P(w_1,\ \ldots,w_n \mid class) = \prod_{i=1}^{n} P(w_i \mid class)$$

Naïve Bayes Representation

- Goal: estimate likelihoods P(document | class) and P(class)
- Likelihood: ***bag of words*** representation
  - The document is a sequence of words (w_1, … , w_n)
  - The order of the words in the document is not important
  - Each word is conditionally independent of the others given document class

$$P(document \mid class) = P(w_1,\ \ldots , w_n \mid class) = \prod_{i=1}^{n} P(w_i \mid class)$$

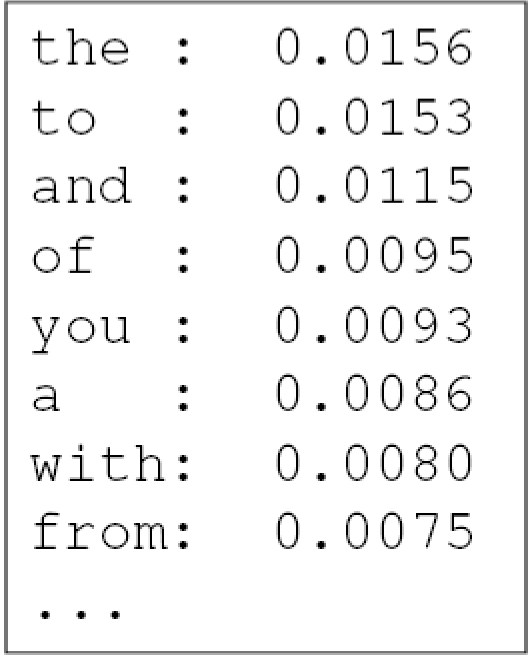
  - Thus, the problem is reduced to estimating marginal likelihoods of individual words P(w_i | class)

Parameter estimation

- Model parameters: feature likelihoods P(word | class) and priors P(class)
  - How do we obtain the values of these parameters?

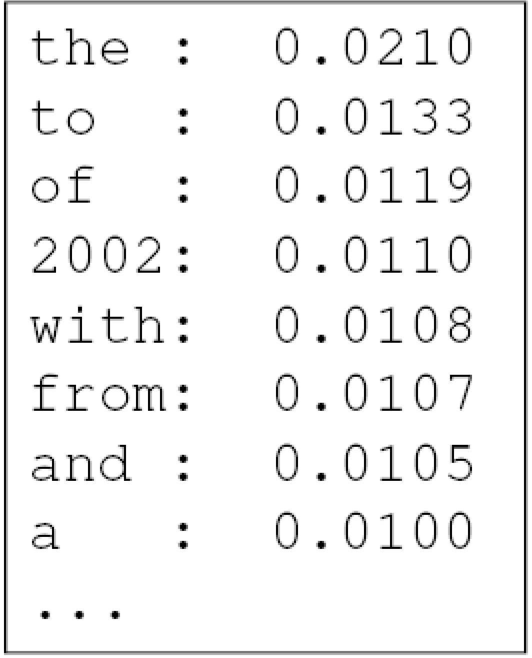
prior

P(word | spam)



the	:	0.0156
to	:	0.0153
and	:	0.0115
of	:	0.0095
you	:	0.0093
a	:	0.0086
with	:	0.0080
from	:	0.0075
...		

P(word | ¬spam)



the	:	0.0210
to	:	0.0133
of	:	0.0119
2002	:	0.0110
with	:	0.0108
from	:	0.0107
and	:	0.0105
a	:	0.0100
...		

Parameter estimation

- Model parameters: feature likelihoods P(word | class) and priors P(class)
  - How do we obtain the values of these parameters?
  - Need *training set* of labeled samples from both classes

$$P(\text{word} \mid \text{class}) = \frac{\text{\# of occurrences of this word in docs from this class}}{\text{total \# of words in docs from this class}}$$

  - This is the *maximum likelihood* (ML) estimate, or estimate that maximizes the likelihood of the training data:

$$\prod_{d=1}^{D} \prod_{i=1}^{n_d} P(w_{d,i} \mid class_{d,i})$$

Parameter estimation

- Parameter estimate:

$$P(\text{word} \mid \text{class}) = \frac{\text{\# of occurrences of this word in docs from this class}}{\text{total \# of words in docs from this class}}$$

- Parameter smoothing: dealing with words that were never seen or seen too few times
  - **Laplacian smoothing:** pretend you have seen every vocabulary word one more time than you actually did

$$P(\text{word} \mid \text{class}) = \frac{\text{\# of occurrences of this word in docs from this class} + 1}{\text{total \# of words in docs from this class} + V}$$

(V: total number of unique words)

Summary: Naïve Bayes for Document Classification

- Assign the document to the class with the highest posterior

$$P(class \mid document)\, \mu\, P(class) \tilde{O}_{i=1}^{n} P(w_i \mid class)$$

- Model parameters:

prior

$$P(\text{class}_1)$$

$$\ldots$$

$$P(\text{class}_K)$$

Likelihood of class 1

$$P(w_1 \mid \text{class}_1)$$

$$P(w_2 \mid \text{class}_1)$$

$$\ldots$$

$$P(w_n \mid \text{class}_1)$$

Likelihood of class K

$$P(w_1 \mid \text{class}_K)$$

$$P(w_2 \mid \text{class}_K)$$

$$\ldots$$

$$P(w_n \mid \text{class}_K)$$

Summary: Naïve Bayes for Document Classification

- Assign the document to the class with the highest posterior

$$P(class \mid document) \propto P(class) \prod_{i=1}^{n} P(w_i \mid class)$$

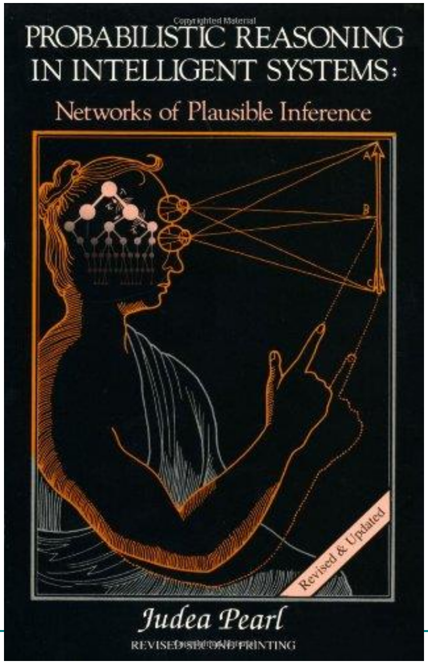
- Note: by convention, one typically works with logs of probabilities instead:

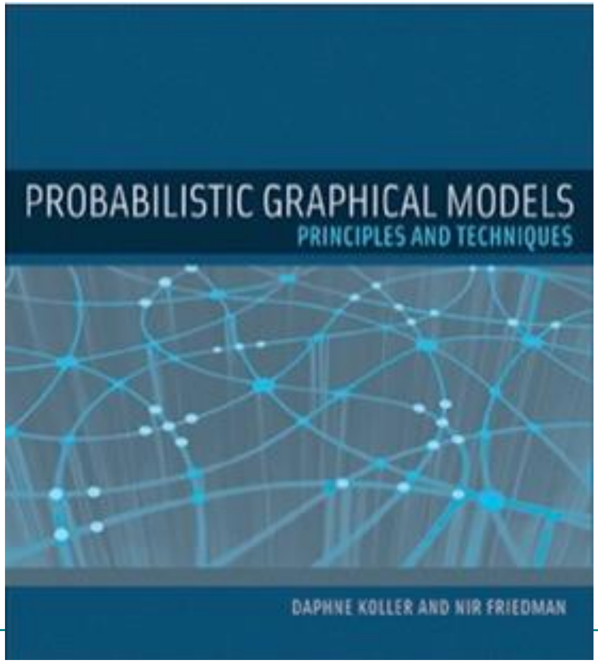
$$L(class \mid document) = \log P(class) + \sum_{i=1}^{n} \log P(w_i \mid class)$$

  – Can help to avoid underflow

Bayesian networks

- More commonly called *graphical models*
- A way to depict conditional independence relationships between random variables
- A compact specification of full joint distributions


Copyrighted Material
PROBABILISTIC REASONING IN INTELLIGENT SYSTEMS:
Networks of Plausible Inference
Revised & Updated
Judea Pearl
REVISED SECOND PRINTING

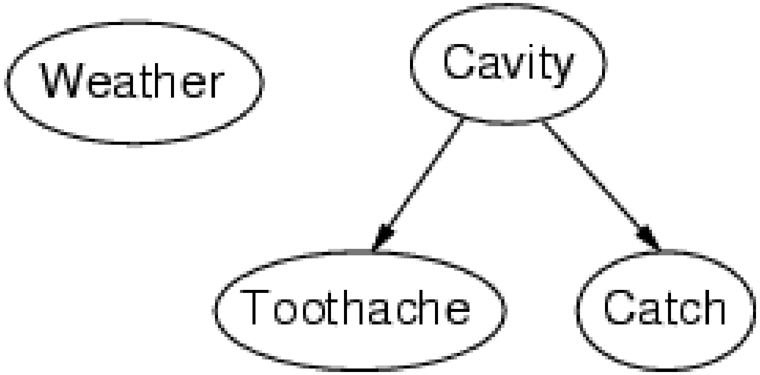

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN

Bayesian networks: Structure

- data structure that represents the dependencies among random variables

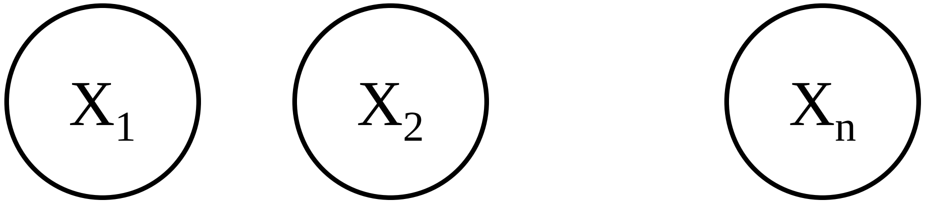
- a directed, *acyclic* graph
- **Nodes:** random variables

- **Arcs:** interactions
- arrow from *X* to *Y* means *X* is a parent of *Y*
- each node *X* has probability distribution **P**(*X* | *Parents*(*X*))



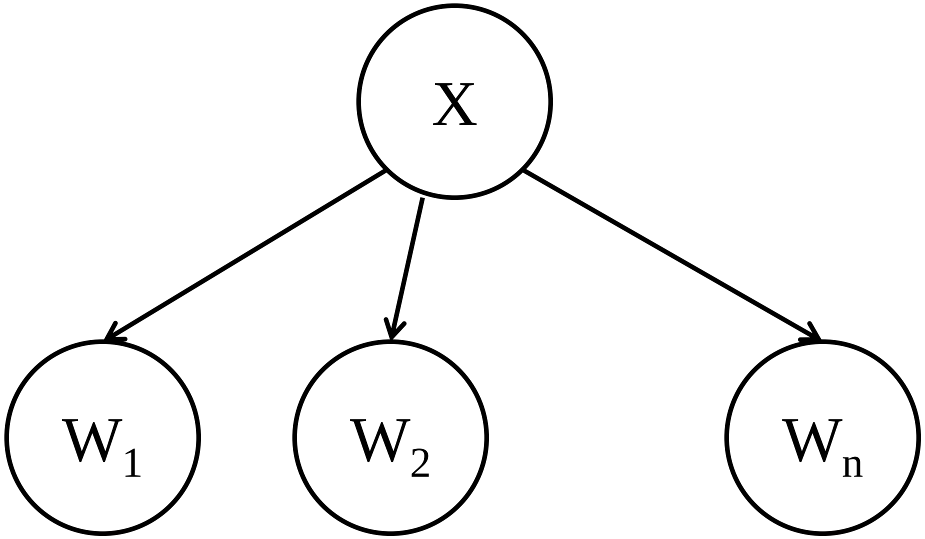
Example: N independent coin flips

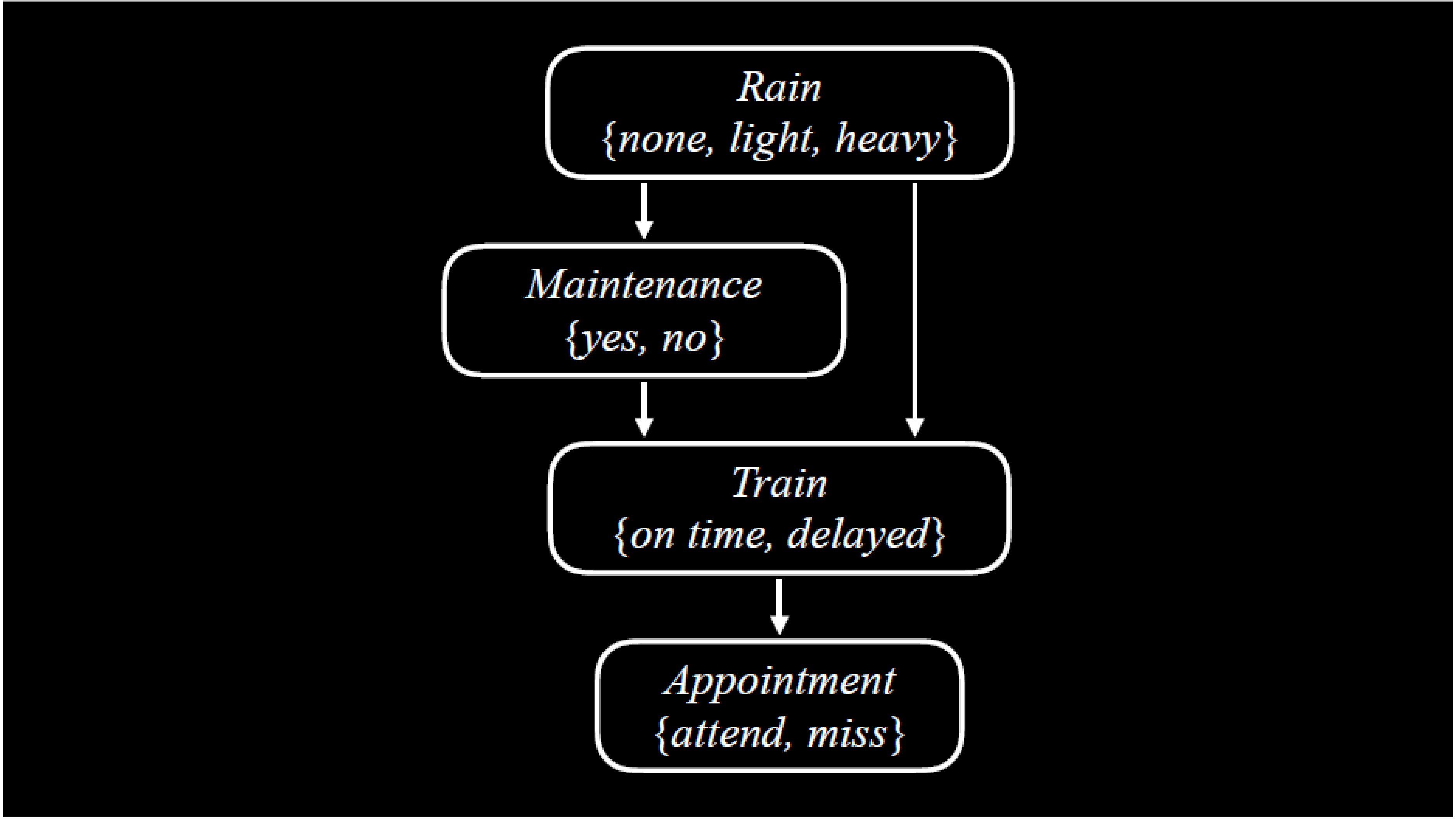
- Complete independence: no interactions



Example: Naïve Bayes document model

- Random variables:
  - X: document class
  - $W_1, \ldots, W_n$: words in the document





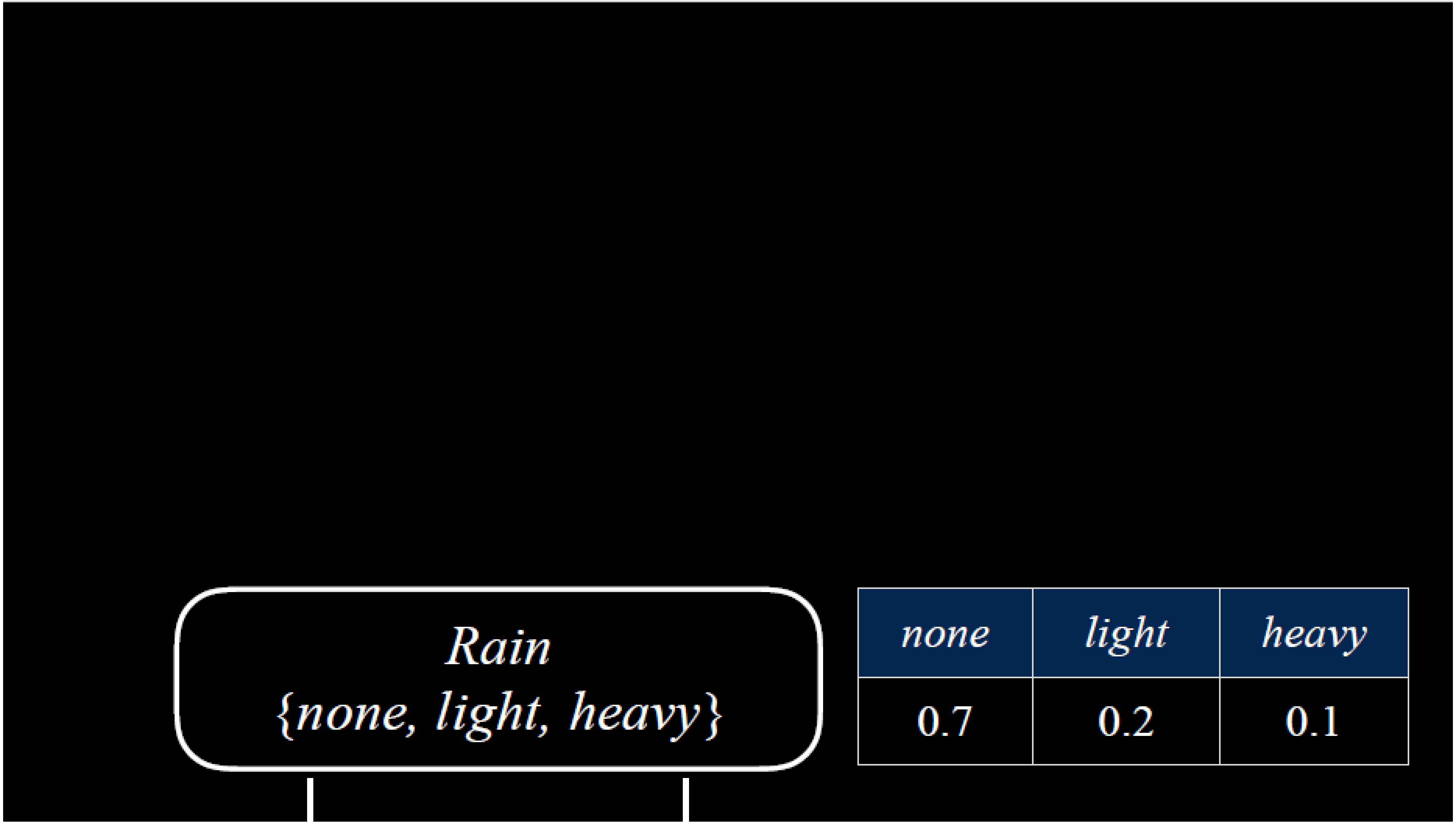
Rain
{none, light, heavy}

Maintenance
{yes, no}

Train
{on time, delayed}

Appointment
{attend, miss}

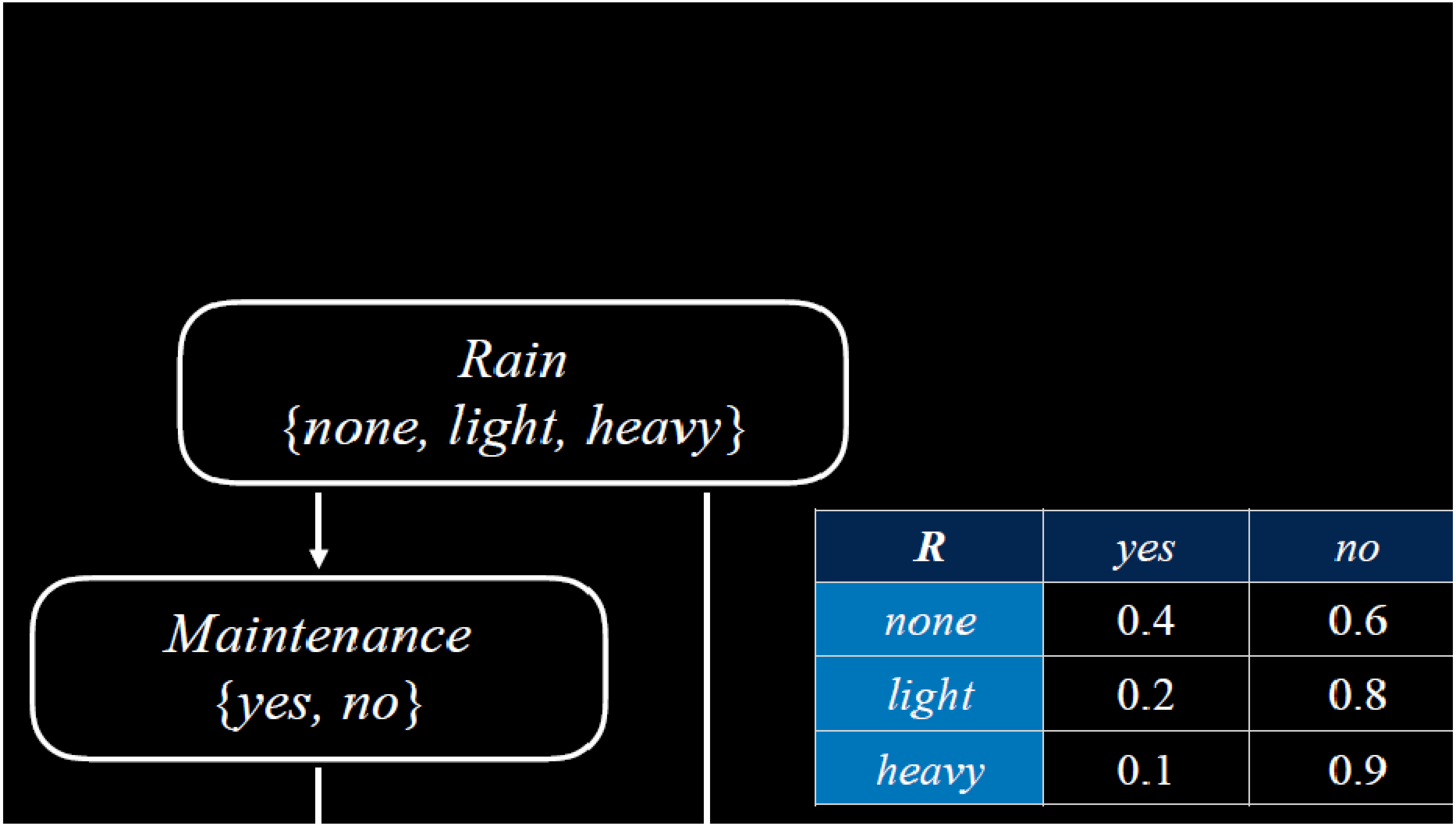
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



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

none	*light*	*heavy*
0.7	0.2	0.1

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

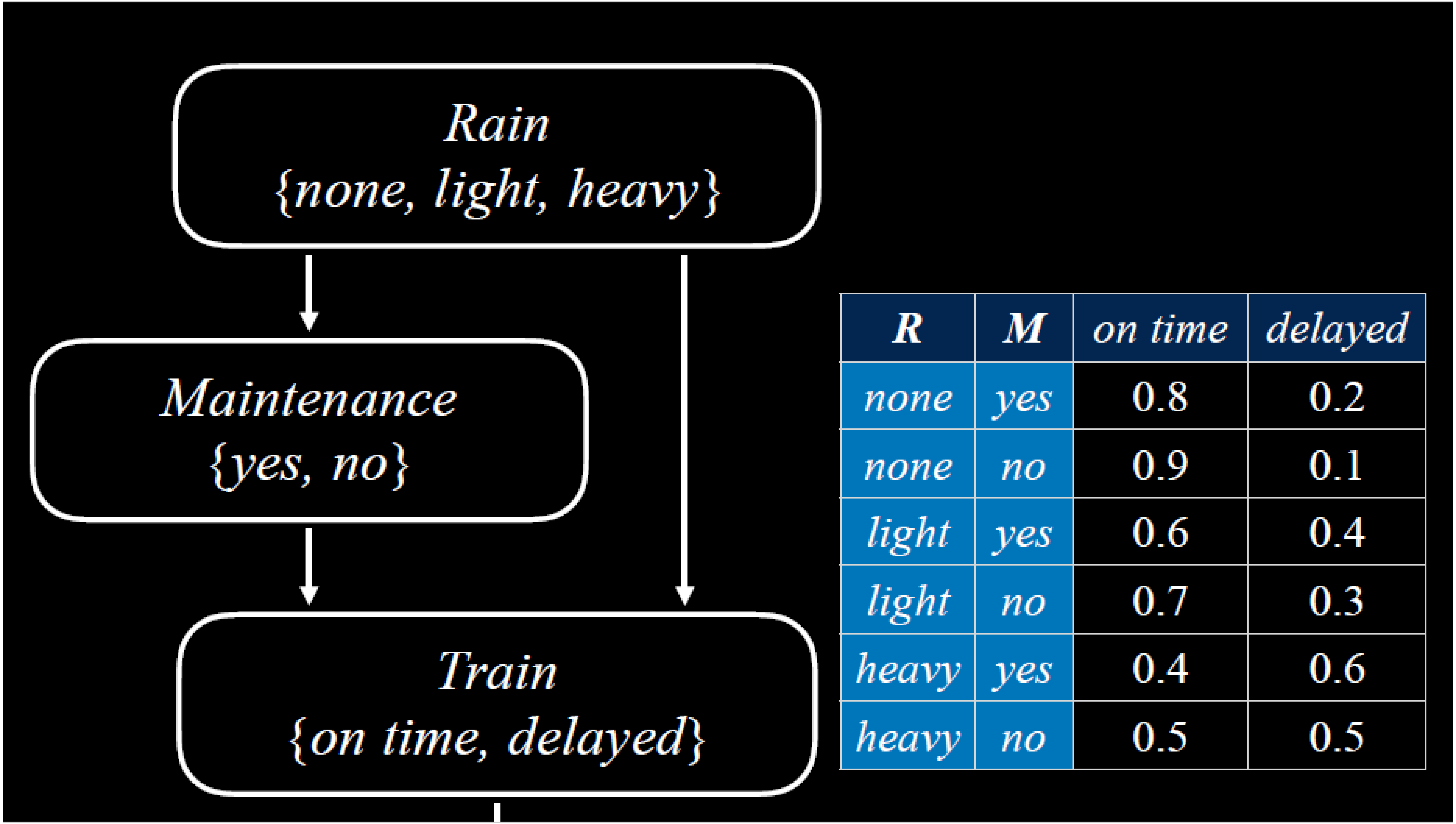


Rain
{*none*, *light*, *heavy*}

Maintenance
{*yes*, *no*}

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



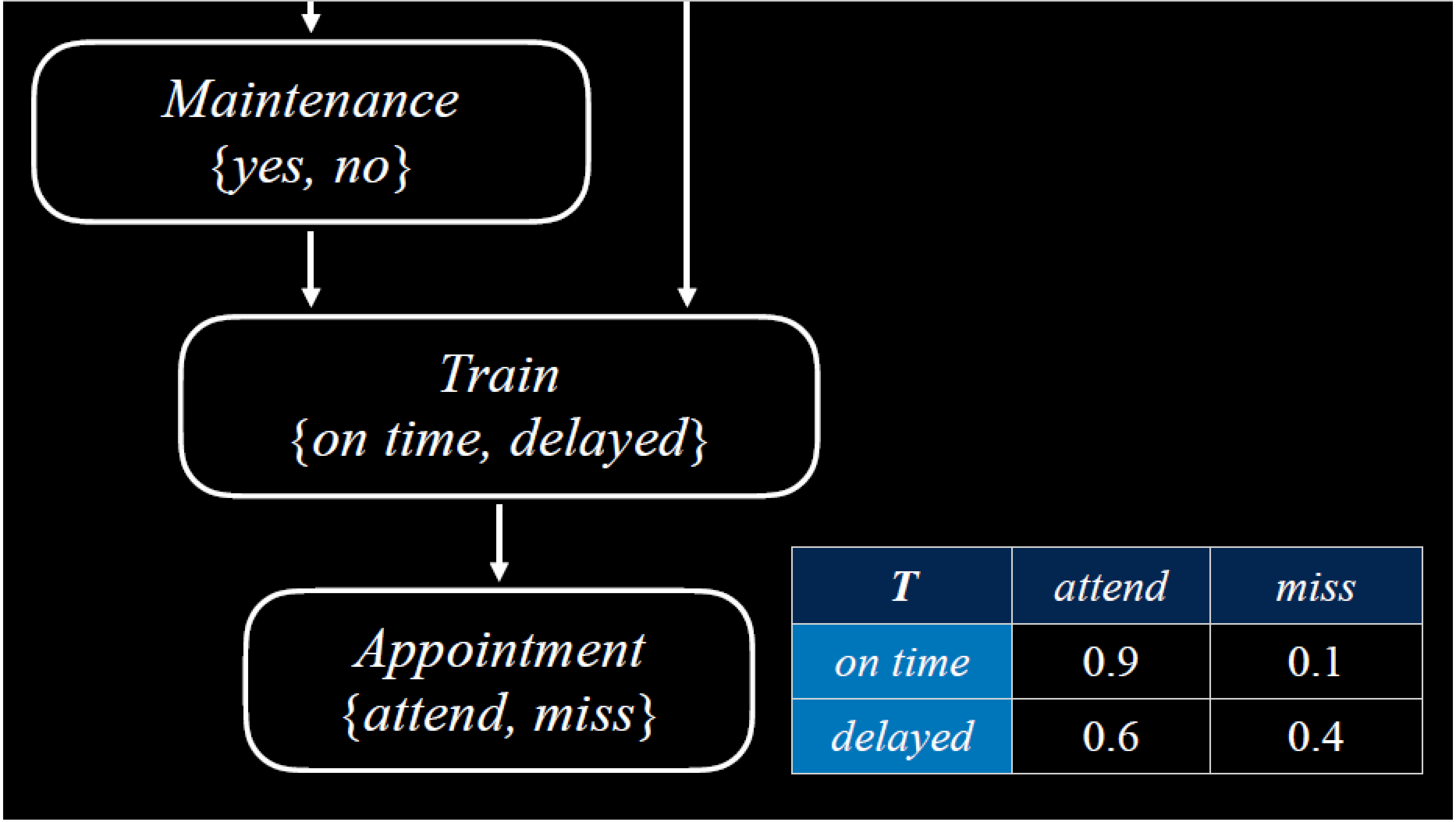
Rain
{none, light, heavy}

Maintenance
{yes, no}

Train
{on time, delayed}

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python,
David J. Malan and Brian Yu



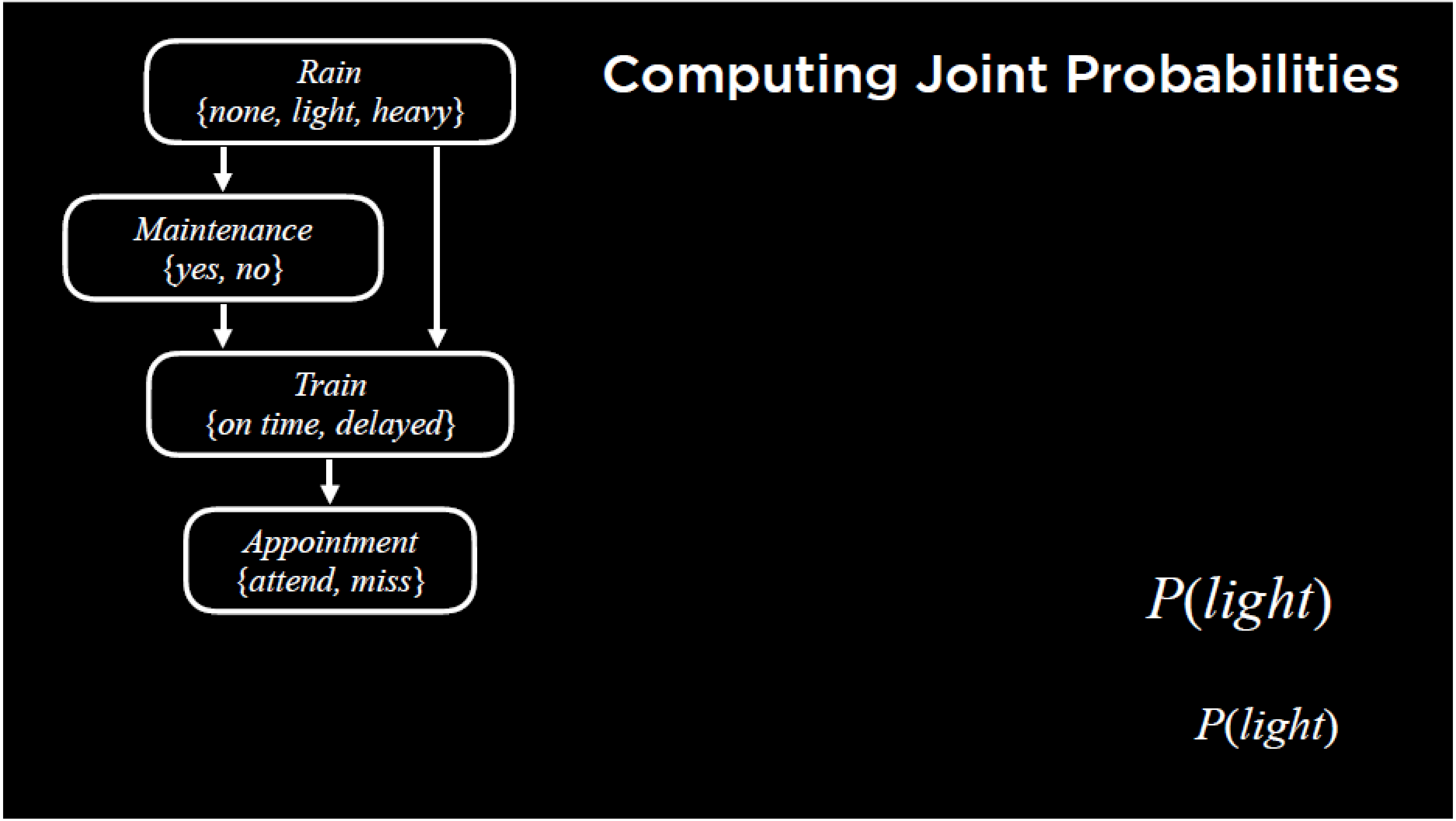
Maintenance
{yes, no}

Train
{on time, delayed}

Appointment
{attend, miss}

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Computing Joint Probabilities

Rain
{*none, light, heavy*}

Maintenance
{*yes, no*}

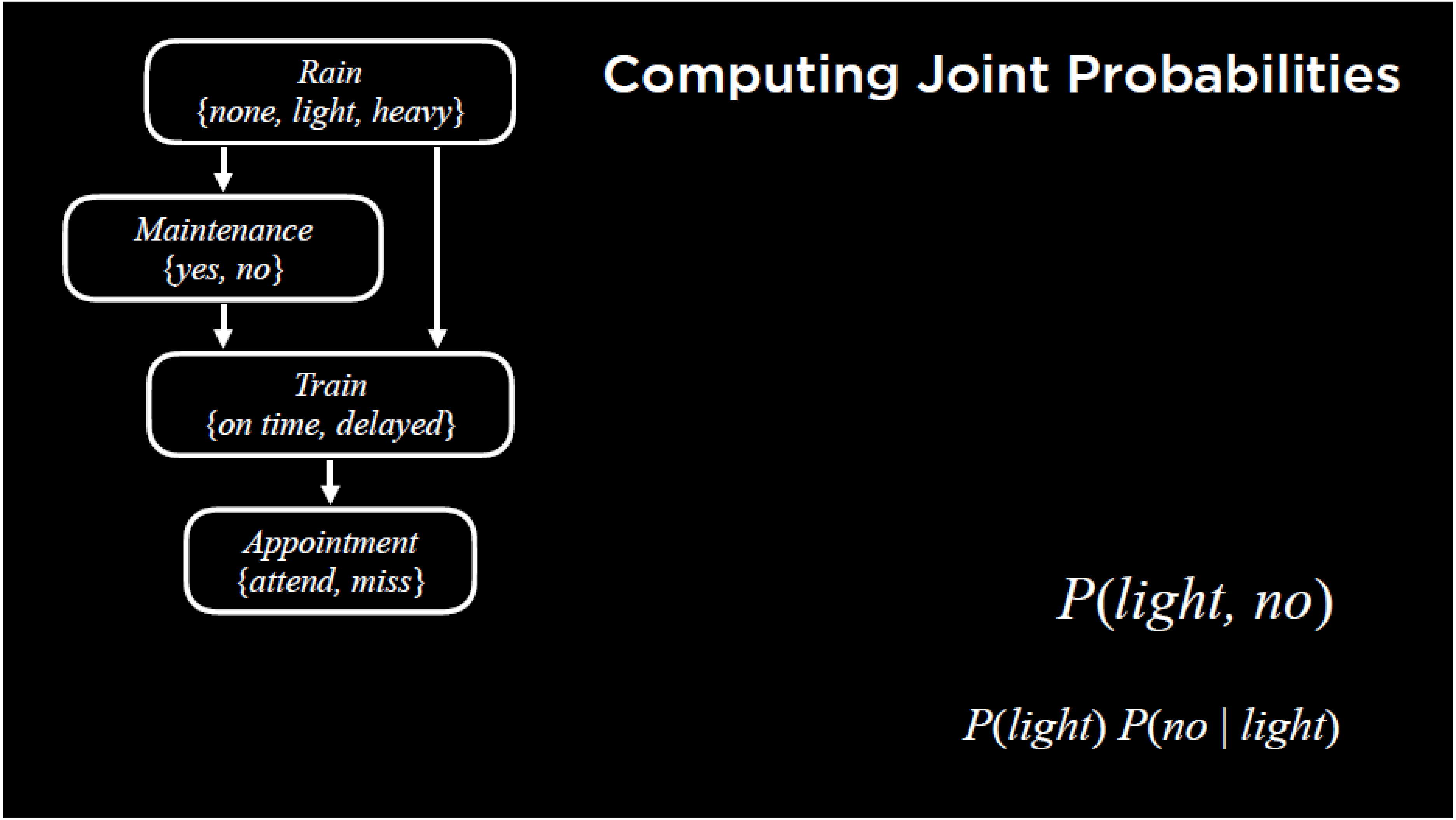
Train
{*on time, delayed*}

Appointment
{*attend, miss*}

$$P(light)$$

$$P(light)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Computing Joint Probabilities

Rain
{none, light, heavy}

Maintenance
{yes, no}

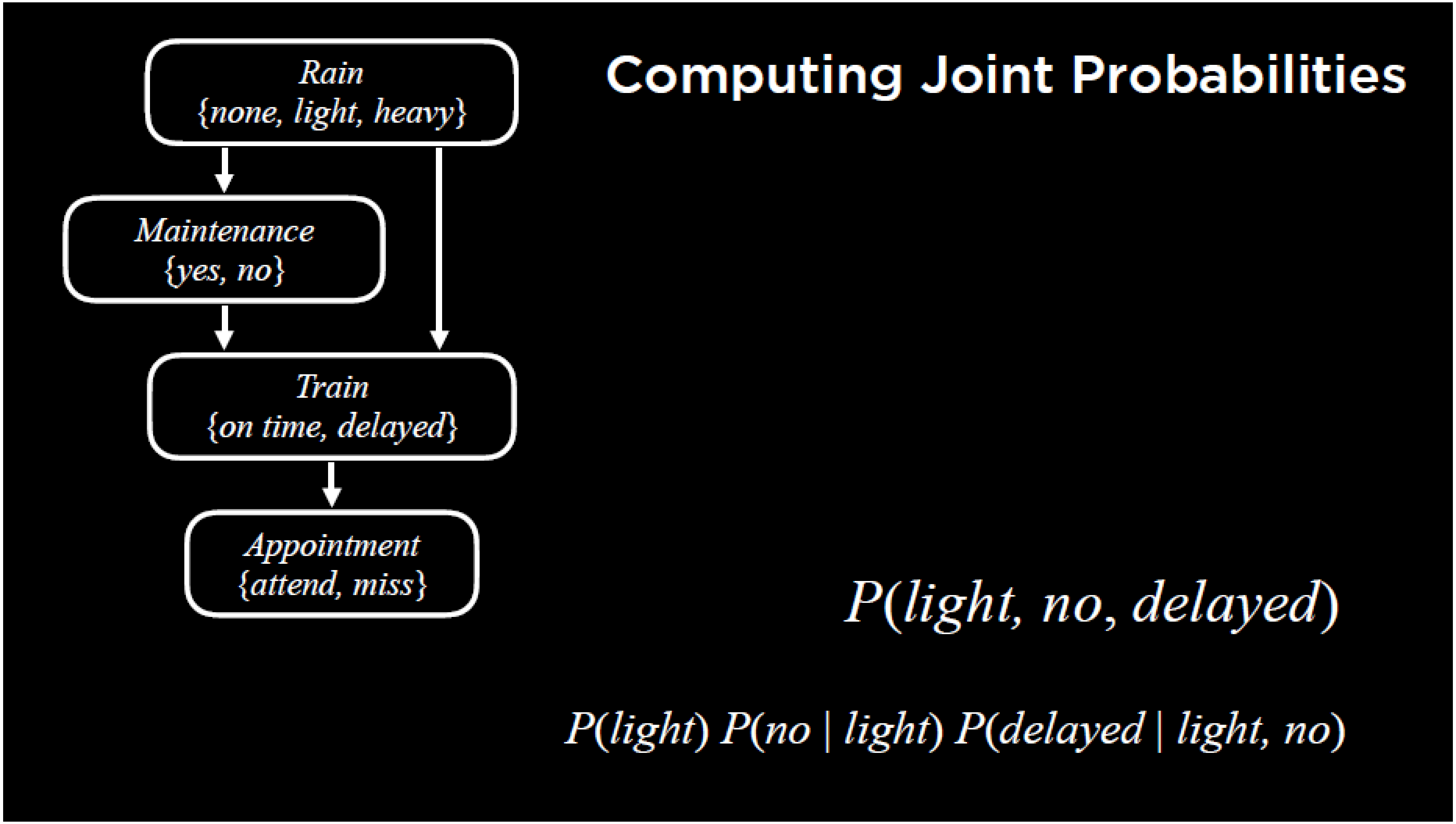
Train
{on time, delayed}

Appointment
{attend, miss}

$$P(light, no)$$

$$P(light)\ P(no \mid light)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Computing Joint Probabilities

Rain
{*none, light, heavy*}

Maintenance
{*yes, no*}

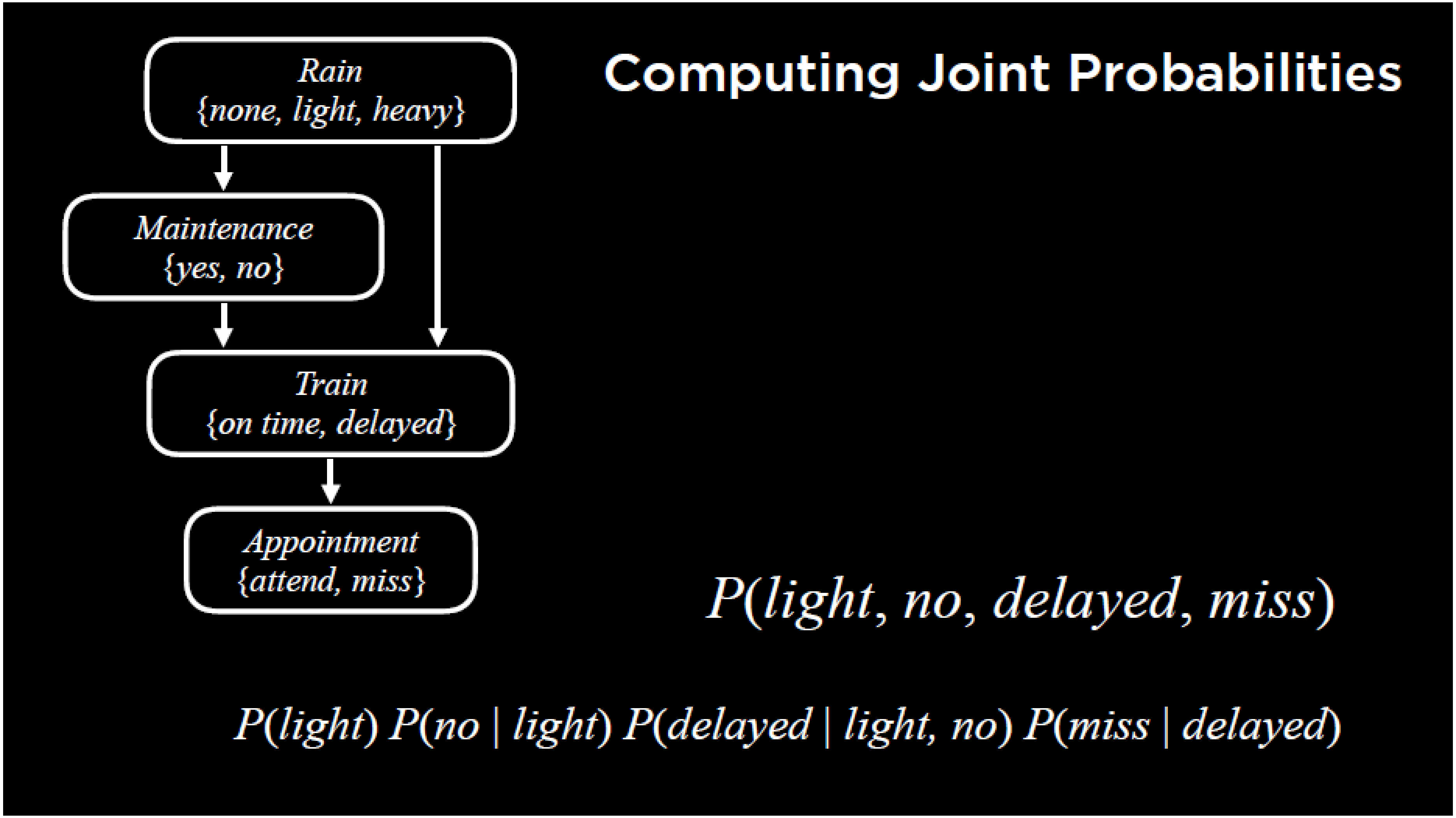
Train
{*on time, delayed*}

Appointment
{*attend, miss*}

$$P(light, no, delayed)$$

$$P(light)\ P(no \mid light)\ P(delayed \mid light, no)$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Computing Joint Probabilities

Rain
{none, light, heavy}

Maintenance
{yes, no}

Train
{on time, delayed}

Appointment
{attend, miss}

$$P(\mathit{light}, \mathit{no}, \mathit{delayed}, \mathit{miss})$$

$$P(\mathit{light})\, P(\mathit{no} \mid \mathit{light})\, P(\mathit{delayed} \mid \mathit{light}, \mathit{no})\, P(\mathit{miss} \mid \mathit{delayed})$$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

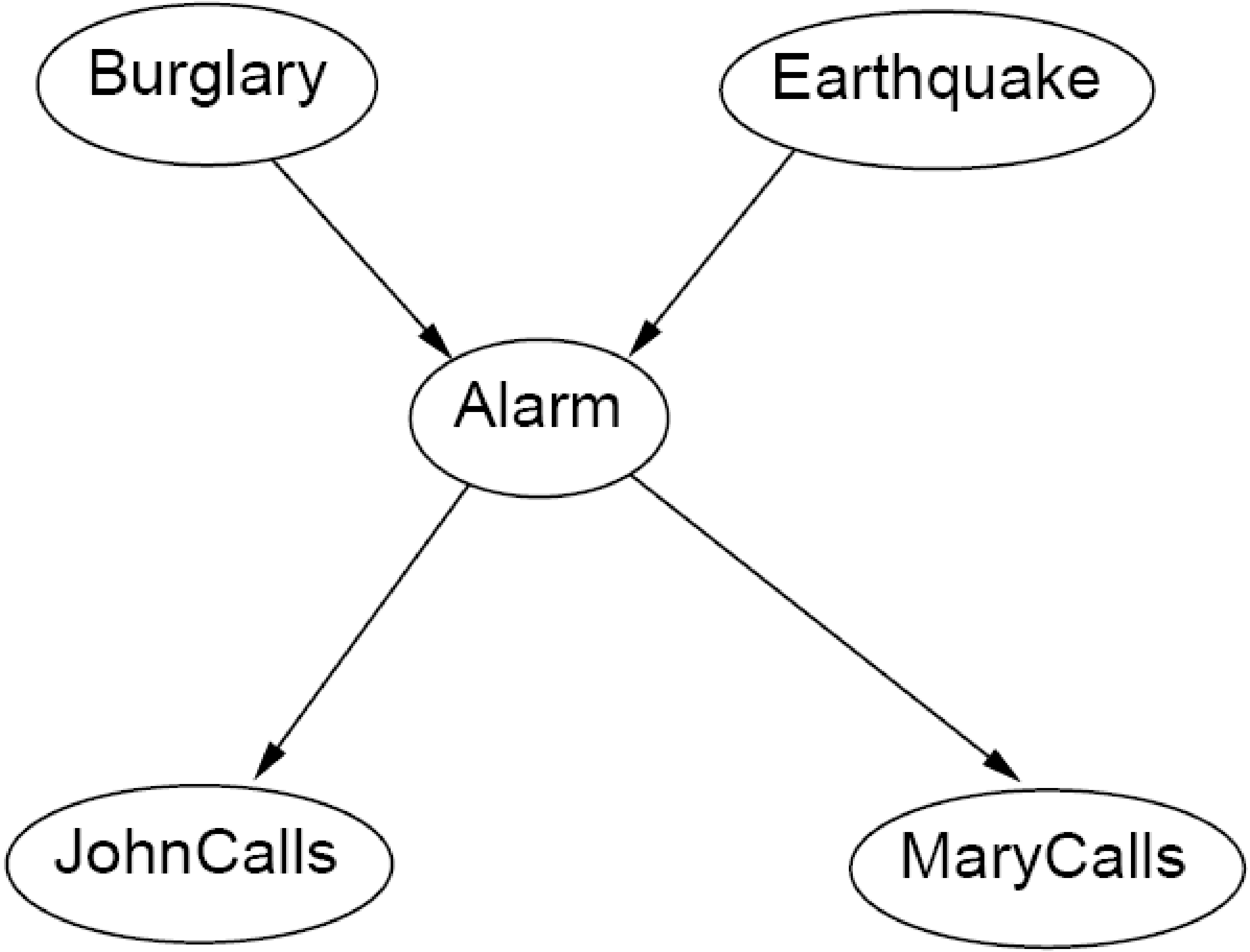
Example: Burglar Alarm

- I have a burglar alarm that is sometimes set off by minor earthquakes. My two neighbors, John and Mary, promised to call me at work if they hear the alarm
- Example inference tasks
  - Suppose Mary calls and John doesn’t call. What is the probability of a burglary?
  - Suppose there is a burglary and no earthquake. What is the probability of John calling?
  - Suppose the alarm went off. What is the probability of burglary?
  - …

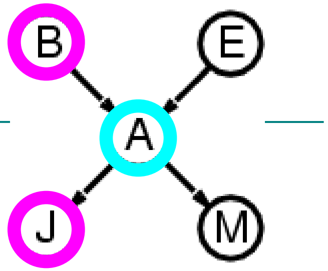
Example: Burglar Alarm

- I have a burglar alarm that is sometimes set off by minor earthquakes. My two neighbors, John and Mary, promised to call me at work if they hear the alarm
- What are the random variables?
  - Burglary, Earthquake, Alarm, John, Mary
- What are the direct influence relationships?
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Example: Burglar Alarm



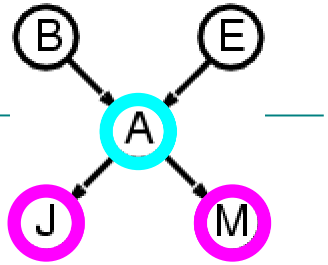
Conditional independence relationships



- Suppose the alarm went off. Does knowing whether there was a burglary change the probability of John calling?

  P(John | Alarm, Burglary) = P(John | Alarm)

Conditional independence relationships



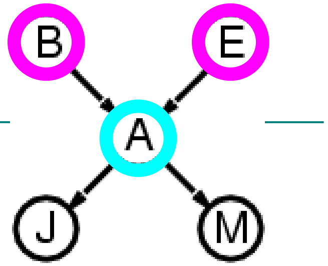
- Suppose the alarm went off. Does knowing whether there was a burglary change the probability of John calling?

  P(John | Alarm, Burglary) = P(John | Alarm)

- Suppose the alarm went off. Does knowing whether John called change the probability of Mary calling?

  P(Mary | Alarm, John) = P(Mary | Alarm)

Conditional independence relationships



- Suppose the alarm went off. Does knowing whether there was a burglary change the probability of John calling?

  P(John | Alarm, Burglary) = P(John | Alarm)

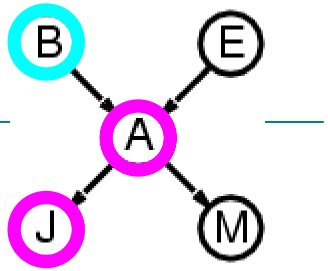
- Suppose the alarm went off. Does knowing whether John called change the probability of Mary calling?

  P(Mary | Alarm, John) = P(Mary | Alarm)

- Suppose the alarm went off. Does knowing whether there was an earthquake change the probability of burglary?

  P(Burglary | Alarm, Earthquake) != P(Burglary | Alarm)

Conditional independence relationships



- Suppose the alarm went off. Does knowing whether there was a burglary change the probability of John calling?

  P(John | Alarm, Burglary) = P(John | Alarm)

- Suppose the alarm went off. Does knowing whether John called change the probability of Mary calling?

  P(Mary | Alarm, John) = P(Mary | Alarm)

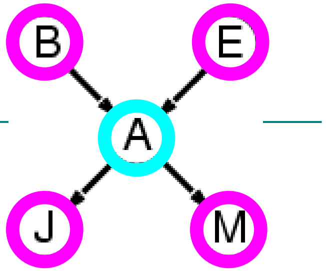
- Suppose the alarm went off. Does knowing whether there was an earthquake change the probability of burglary?

  P(Burglary | Alarm, Earthquake) != P(Burglary | Alarm)

- Suppose there was a burglary. Does knowing whether John called change the probability that the alarm went off?

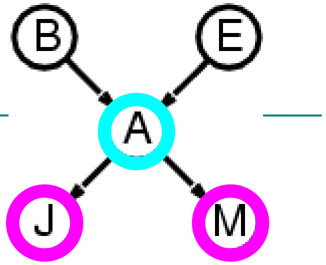
  P(Alarm | Burglary, John) != P(Alarm | Burglary)

Conditional independence relationships



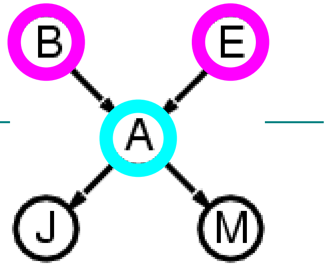
- John and Mary are conditionally independent of Burglary and Earthquake given Alarm
    - *Children* are conditionally independent of *ancestors* given *parents*

Conditional independence relationships



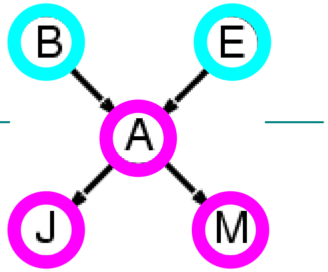
- John and Mary are conditionally independent of Burglary and Earthquake given Alarm
  - *Children* are conditionally independent of *ancestors* given *parents*
- John and Mary are conditionally independent of each other given Alarm
  - *Siblings* are conditionally independent of each other given *parents*

Conditional independence relationships



- John and Mary are conditionally independent of Burglary and Earthquake given Alarm
  - *Children* are conditionally independent of *ancestors* given *parents*
- John and Mary are conditionally independent of each other given Alarm
  - *Siblings* are conditionally independent of each other given *parents*
- Burglary and Earthquake are *not* conditionally independent of each other given Alarm
  - *Parents* are *not* conditionally independent given *children*

Conditional independence relationships



- John and Mary are conditionally independent of Burglary and Earthquake given Alarm
  - *Children* are conditionally independent of *ancestors* given *parents*
- John and Mary are conditionally independent of each other given Alarm
  - *Siblings* are conditionally independent of each other given *parents*
- Burglary and Earthquake are *not* conditionally independent of each other given Alarm
  - *Parents* are *not* conditionally independent given *children*
- Alarm is *not* conditionally independent of John and Mary given Burglary and Earthquake
  - Nodes are *not* conditionally independent of *children* given *parents*

- **General rule:** each node is conditionally independent of its *non-descendants* given its *parents*

Conditional independence and the joint distribution

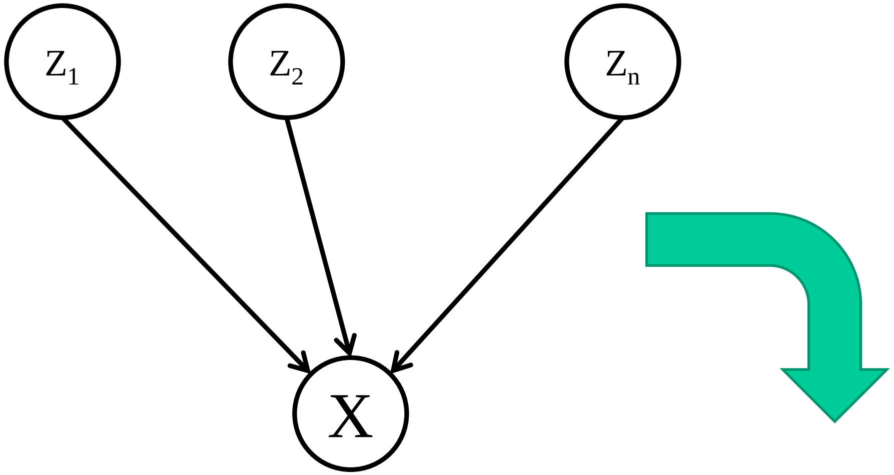
- **General rule:** each node is conditionally independent of its *non-descendants* given its *parents*
- Suppose the nodes $X_1, \ldots, X_n$ are sorted in topological order (parents before children)
- To get the joint distribution $P(X_1, \ldots, X_n)$, use chain rule:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid X_1, \ldots, X_{i-1}\right)$$

$$= \prod_{i=1}^{n} P\left(X_i \mid Parents(X_i)\right)$$

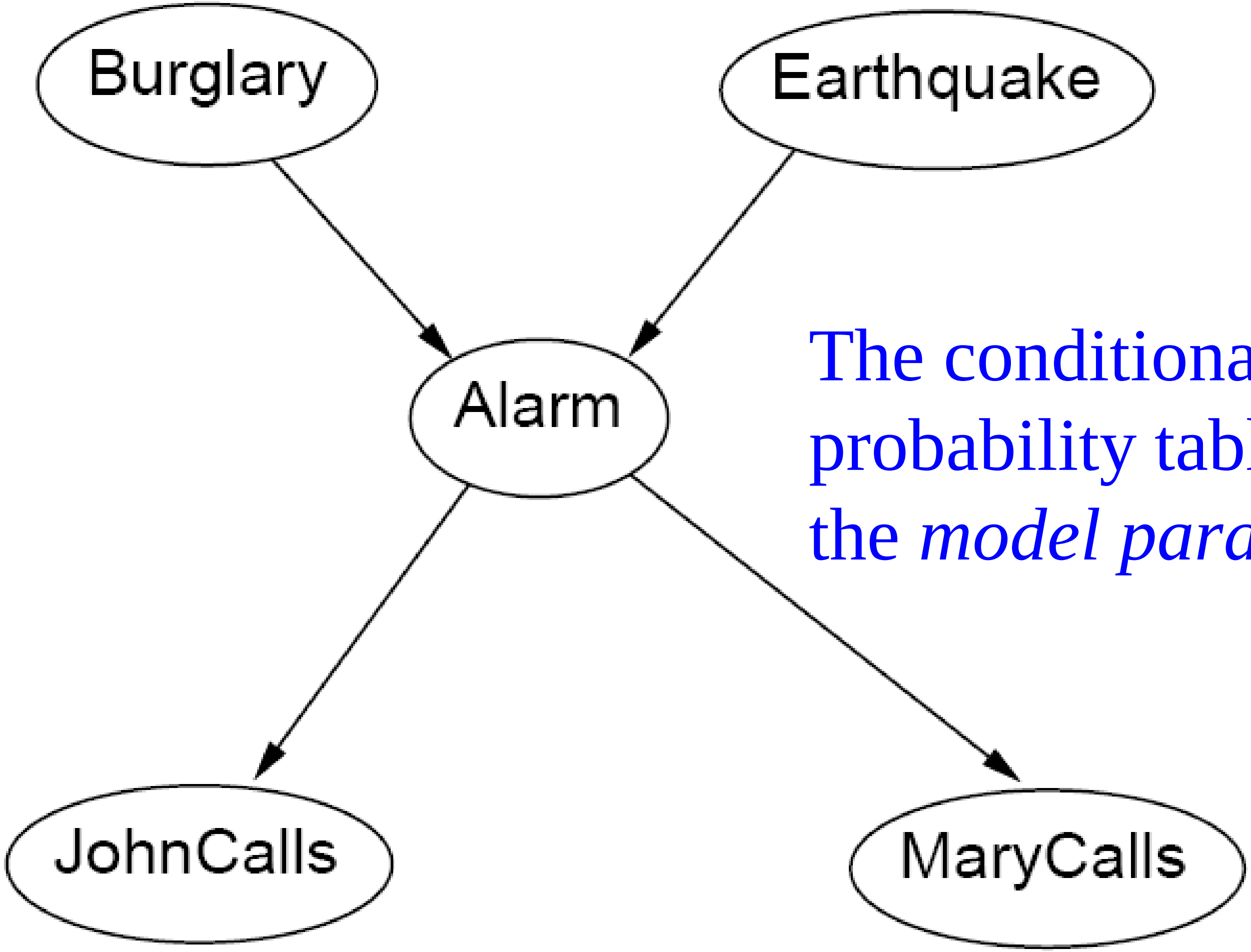
Conditional probability distributions

- To specify the full joint distribution, we need to specify a *conditional* distribution for each node given its parents: $P\ (X \mid \text{Parents}(X))$



$P\ (X \mid Z_1, \ldots, Z_n)$

Example: Burglar Alarm

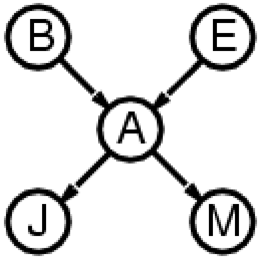


The conditional probability tables are the *model parameters*

The joint probability distribution

$$\boxed{P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P\left(X_i \mid Parents(X_i)\right)}$$

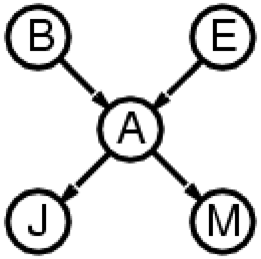
- For example, P(j, m, a, ¬b, ¬e)
  = P(¬b) P(¬e) P(a | ¬b, ¬e) P(j | a) P(m | a)



Compactness

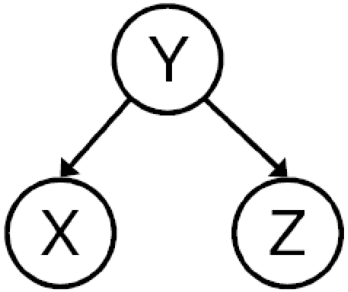
- Suppose we have a Boolean variable X_i with k Boolean parents. How many rows does its conditional probability table have?
  - 2^k rows for all the combinations of parent values
  - Each row requires one number for $P(X_i = \text{true} \mid \text{parent values})$
- If each variable has no more than k parents, how many numbers does the complete network require?
  - $O(n \cdot 2^k)$ numbers – vs. $O(2^n)$ for the full joint distribution
- How many nodes for the burglary network?

  $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. 2^5-1 = 31)



Conditional independence

- Common cause



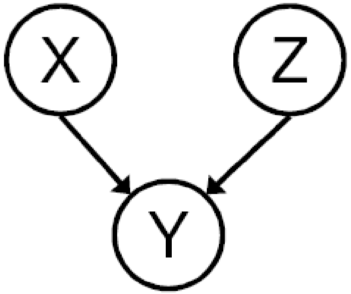
Y: Project due

X: Newsgroup busy

Z: Lab full

- Are X and Z independent?
  - No
- Are they conditionally independent given Y?
  - Yes

- Common effect



X: Raining

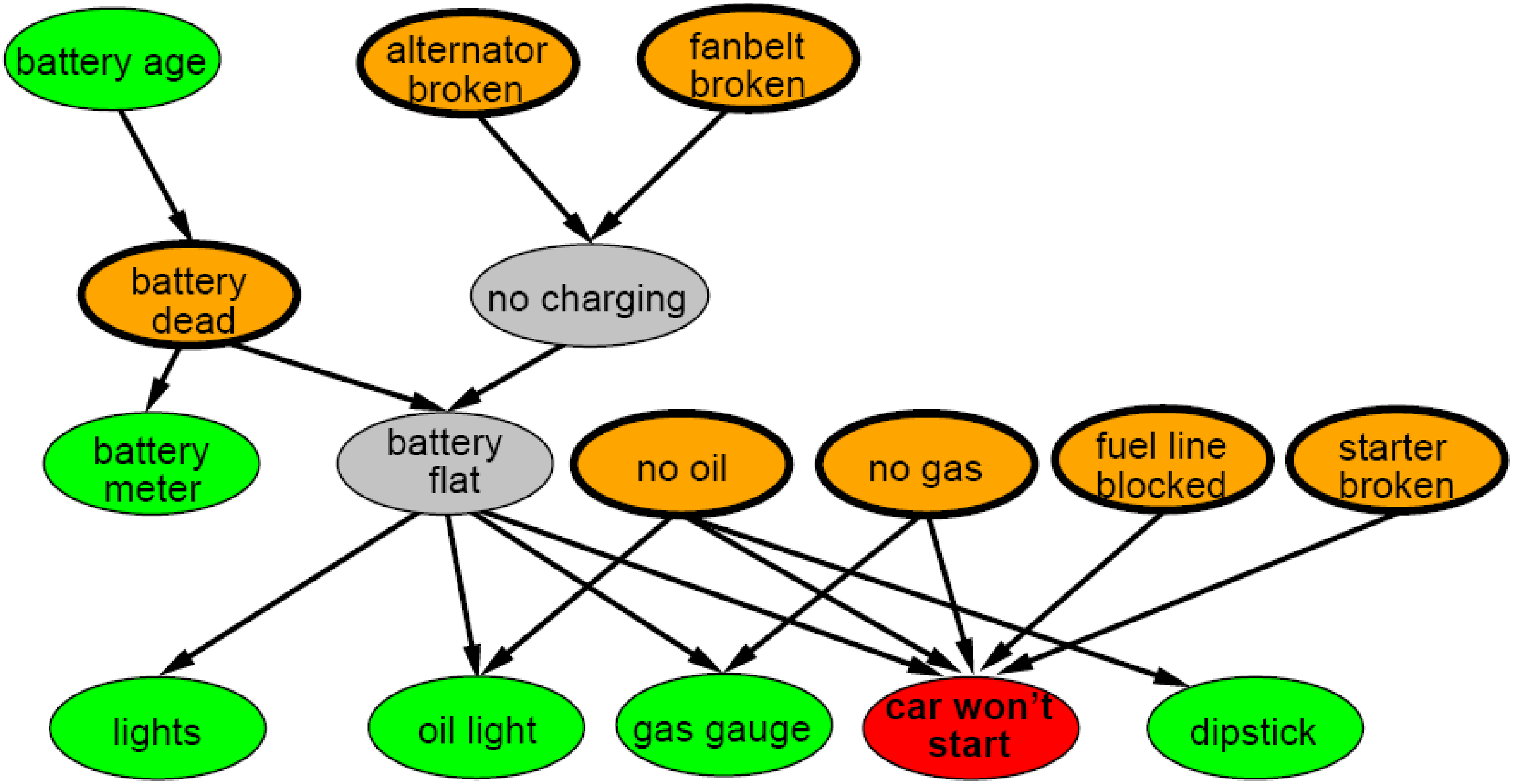
Z: Ballgame

Y: Traffic

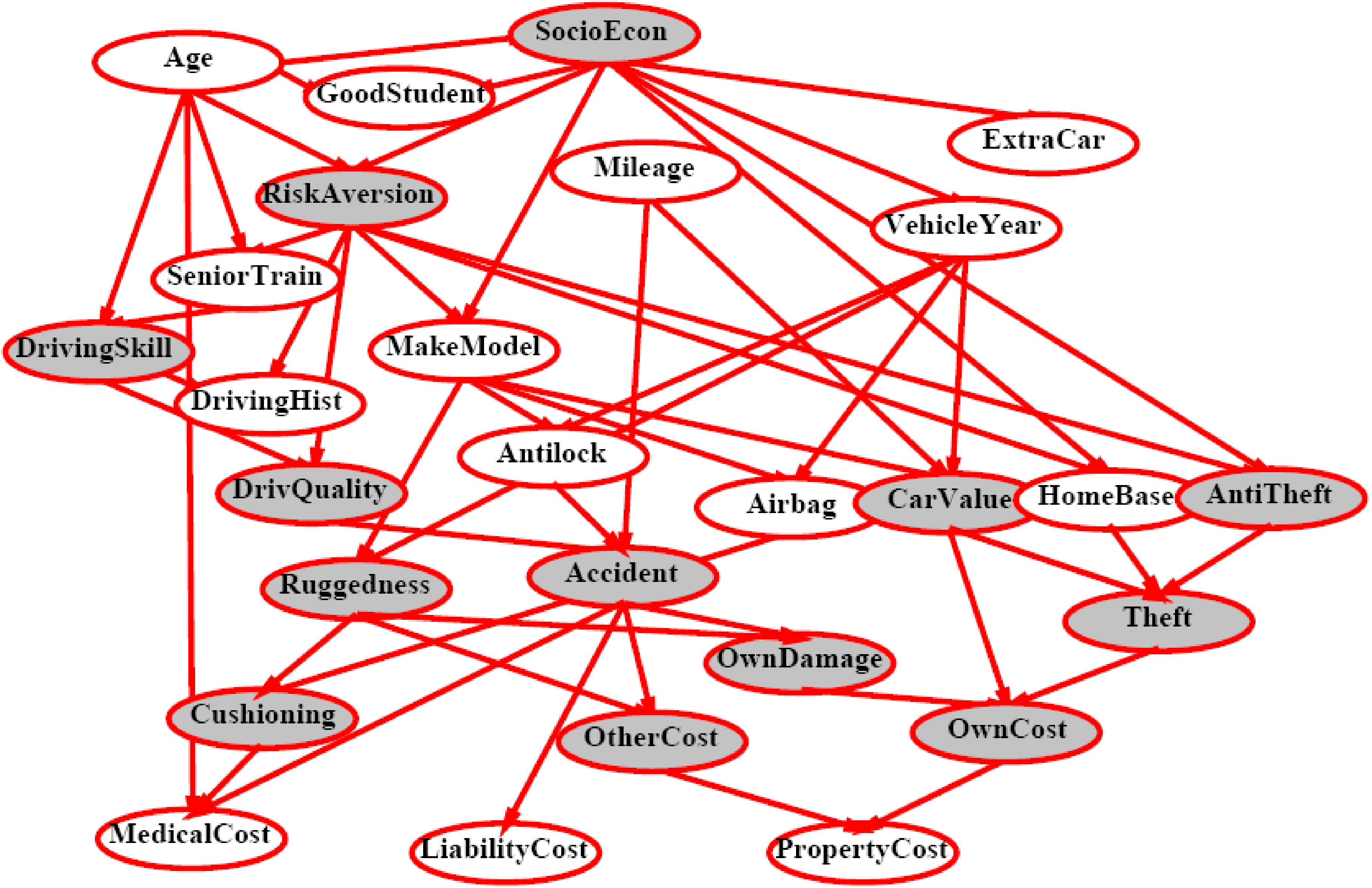
- Are X and Z independent?
  - Yes
- Are they conditionally independent given Y?
  - No

A more realistic Bayes Network: Car diagnosis

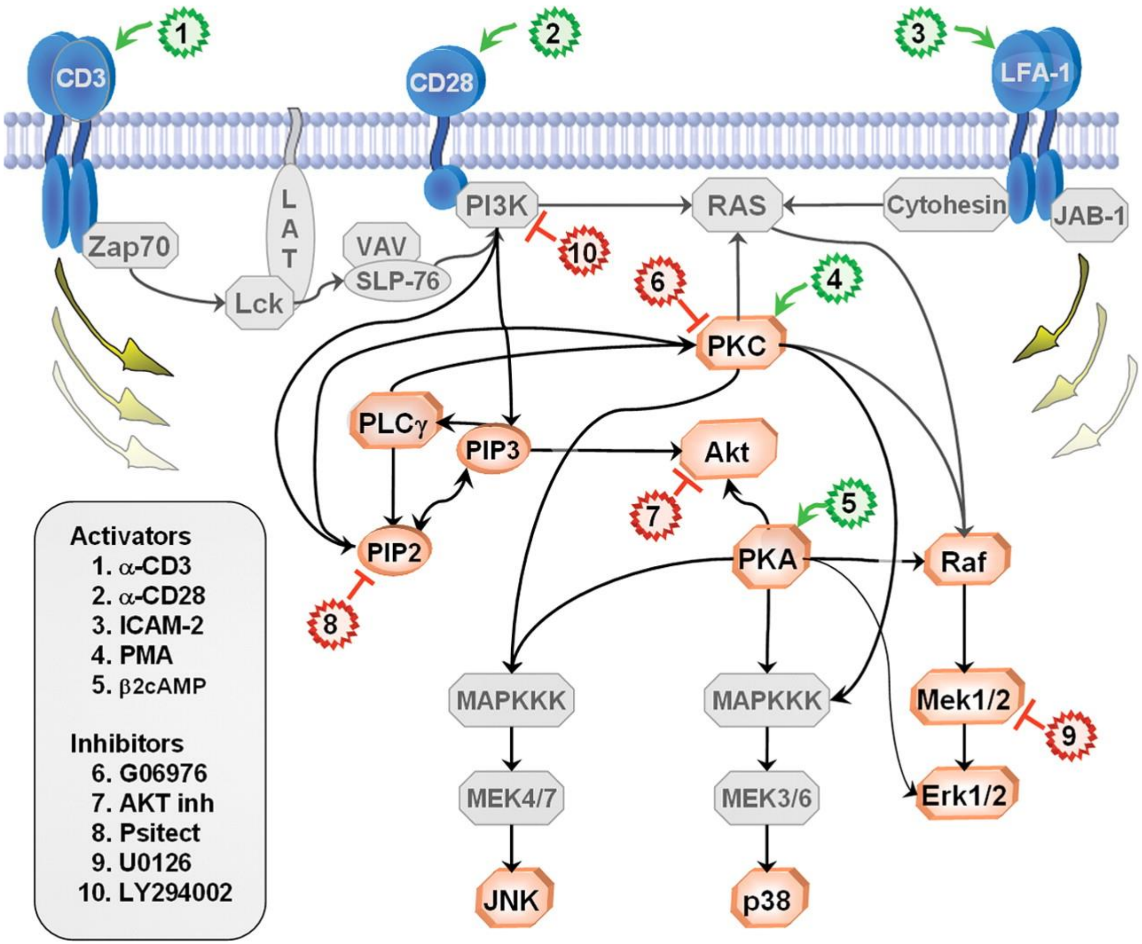
- Initial observation: car won’t start
- Orange: “broken, so fix it” nodes
- Green: testable evidence
- Gray: “hidden variables” to ensure sparse structure, reduce parameteres



Car insurance



In research literature…



Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan

(22 April 2005) *Science* **308** (5721), 523.

In research literature…

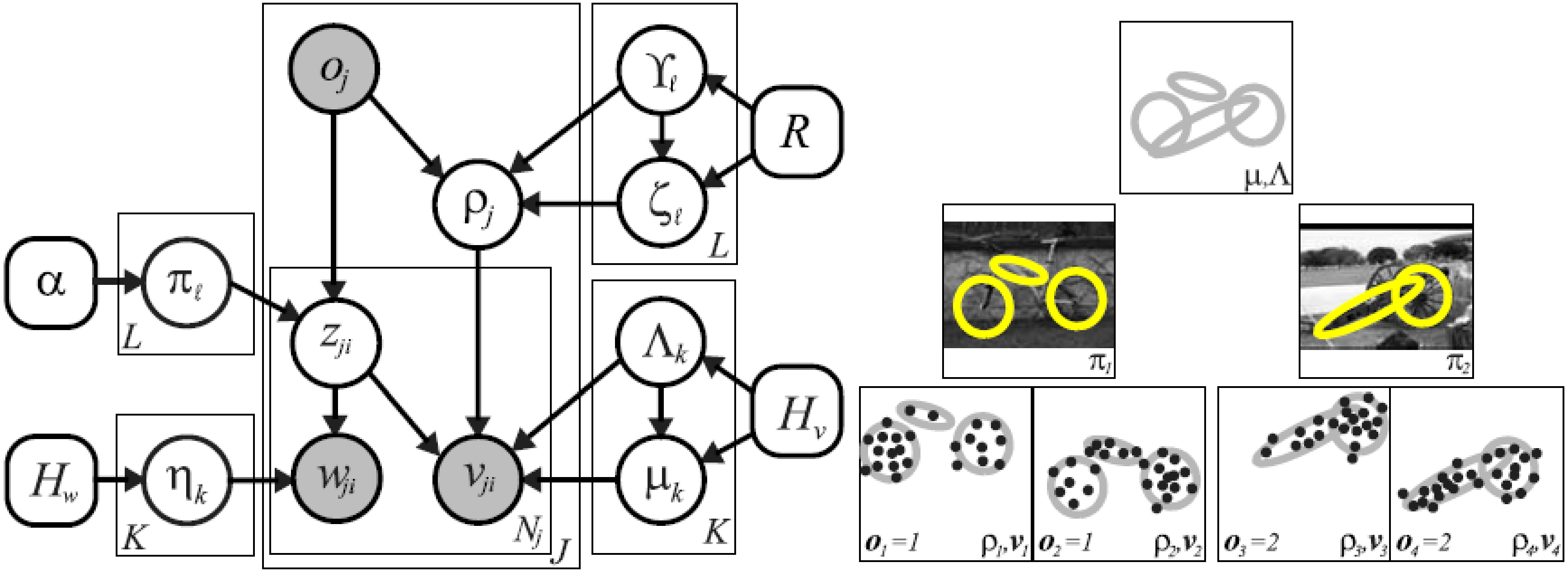


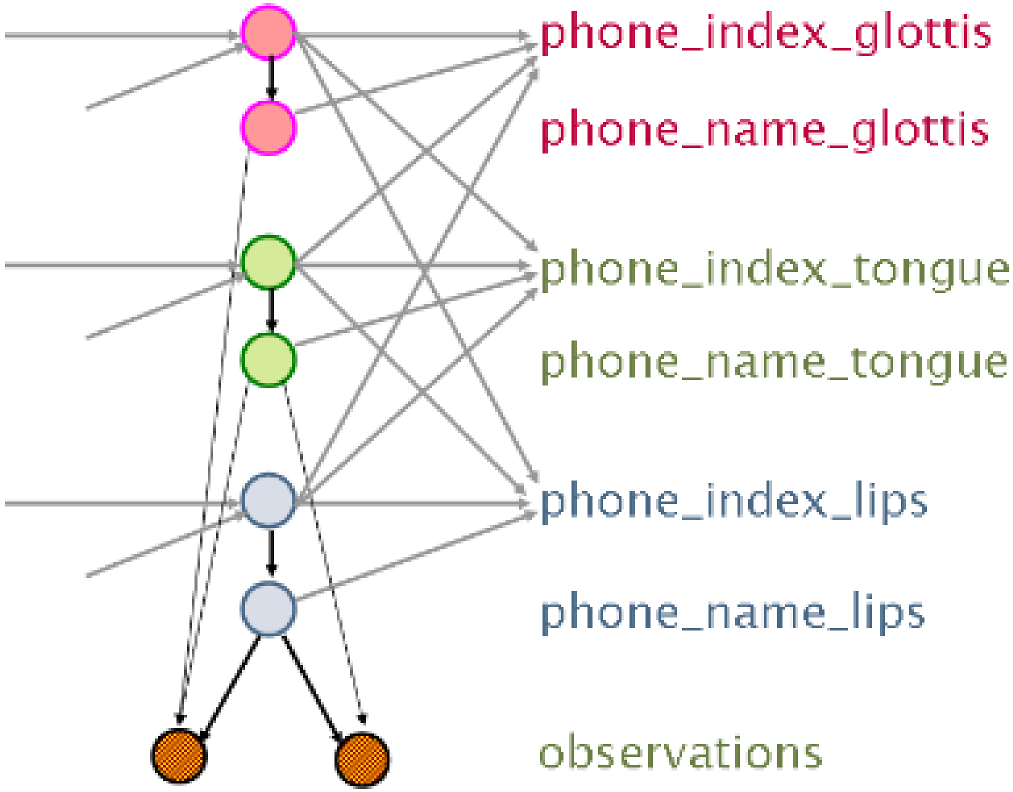
Fig. 3 A parametric, fixed-order model which describes the visual appearance of L object categories via a common set of K shared parts. The j^{th} image depicts an instance of object category o_j, whose position is determined by the reference transformation ρ_j. The appearance w_{ji} and position v_{ji}, relative to ρ_j, of visual features are determined by assignments $z_{ji} \sim \pi_{o_j}$ to latent parts. The cartoon example illustrates how a wheel part might be shared among two categories, *bicycle* and *cannon*. We show feature positions (but not appearance) for two hypothetical samples from each category

Describing Visual Scenes Using Transformed Objects and Parts

E. Sudderth, A. Torralba, W. T. Freeman, and A. Willsky.

International Journal of Computer Vision, No. 1-3, May 2008, pp. 291-330.

In research literature…

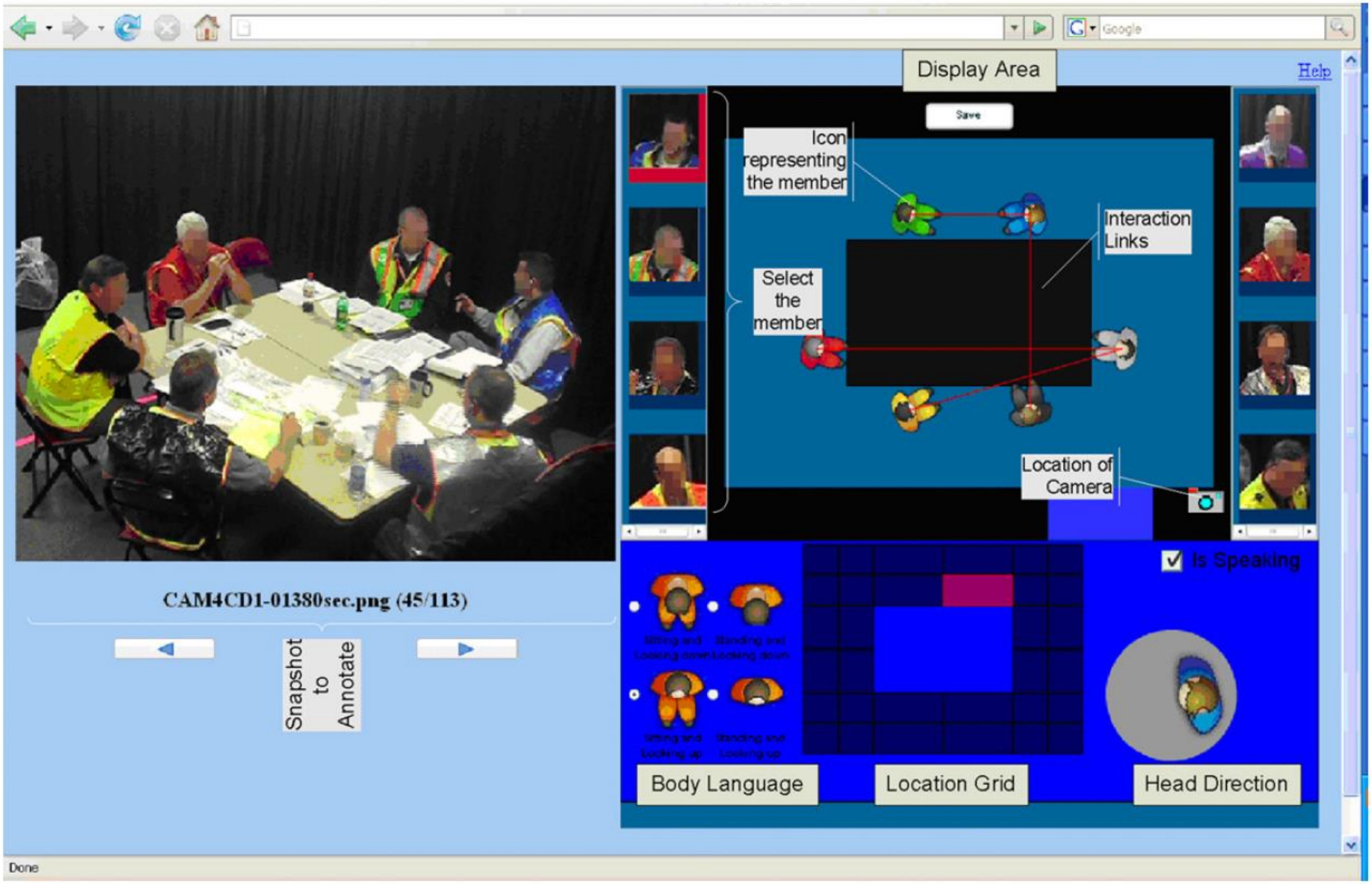


Audiovisual Speech Recognition with Articulator Positions as Hidden Variables

Mark Hasegawa-Johnson, Karen Livescu, Partha Lal and Kate Saenko

International Congress on Phonetic Sciences 1719:299-302, 2007

In research literature…

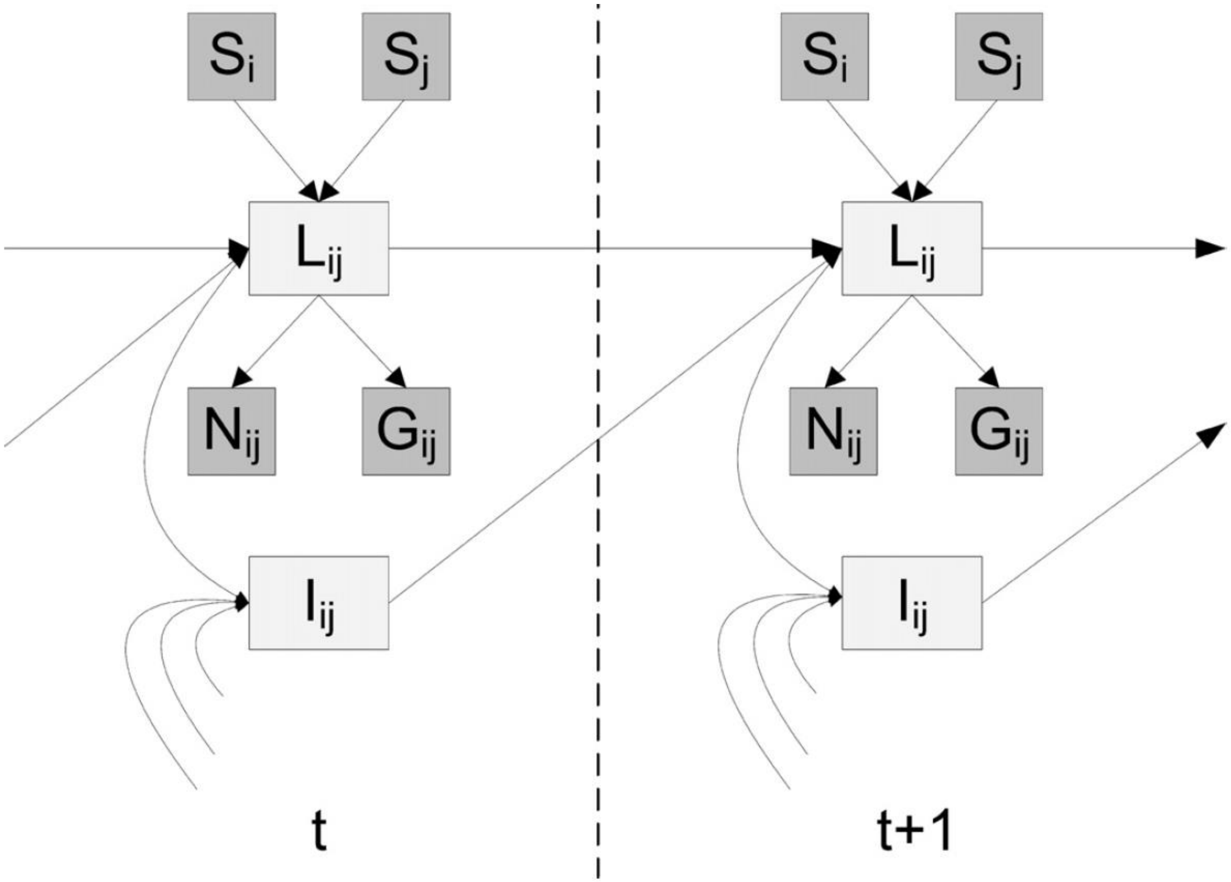


Detecting interaction links in a collaborating group using manually annotated data

S. Mathur, M.S. Poole, F. Pena-Mora, M. Hasegawa-Johnson, N. Contractor

Social Networks 10.1016/j.socnet.2012.04.002

In research literature…



Detecting interaction links in a collaborating group using manually annotated data

S. Mathur, M.S. Poole, F. Pena-Mora, M. Hasegawa-Johnson, N. Contractor

Social Networks 10.1016/j.socnet.2012.04.002

Summary

- Bayesian networks provide a natural representation for (causally induced) conditional independence
- Topology + conditional probability tables
- Generally easy for domain experts to construct

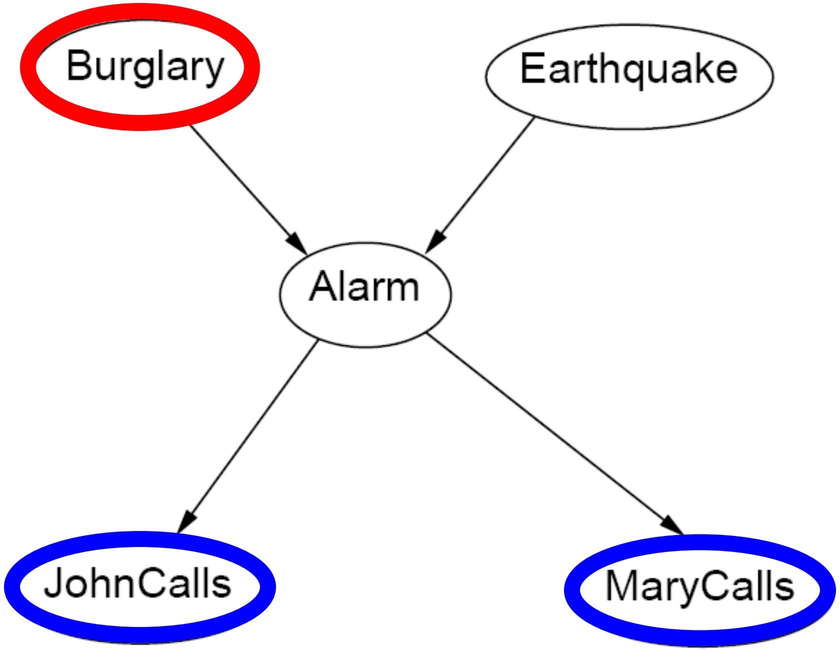
Bayes network inference

- **A general scenario:**
  - Query *variables:* **X**
  - variable for which to compute distribution
  - *Evidence* variables and their values: **E = e**
  - observed variables for event **e**
  - *Unobserved/Hidden* variables: **Y**
  - non-evidence, non-query variable.
- **Inference problem**: answer questions about the query variables given the evidence variables

- Goal: Calculate **P**(X | **e**)

Bayes network inference

- **A general scenario:**
  - Query *variables:* **X**
  - *Evidence* (*observed*) variables and their values: **E = e**
  - *Unobserved* variables: **Y**
- **Inference problem**: answer questions about the query variables given the evidence variables
- **Example:** what is the probability of a burglary given that John and Mary called?



Bayes network inference

- **A general scenario:**
  - Query *variables:* **X**
  - *Evidence* (*observed*) variables and their values: **E** = **e**
  - *Unobserved* variables: **Y**
- **Inference problem**: answer questions about the query variables given the evidence variables
  - This can be done using the posterior distribution P(**X** | **E** = **e**)

$$P(\boldsymbol{X} \mid \boldsymbol{E} = \boldsymbol{e})$$

  - The posterior can be derived from the full joint P(**X**, **E**, **Y**)
- Since Bayesian networks can afford exponential savings in representing joint distributions, can they afford similar savings for inference?

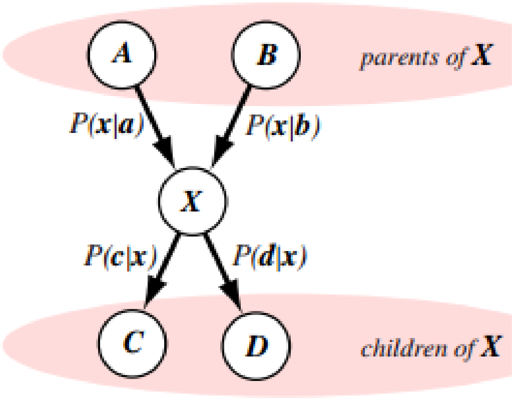
Full Joint distribution

- Consider a Bayes network with n variables $x_1, \ldots, x_n$.
- Denote the parents of a node x_i as $\mathcal{P}(x_i)$.
- Then, we can decompose the joint distribution into the product of conditionals

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P\left(x_i | \mathcal{P}(x_i)\right) \qquad ($$

$$\begin{aligned}
\mathbf{P}(x_1, \ldots, x_n) &= P(x_n | x_{n-1}, \ldots x_1) P(x_{n-1}, \ldots x_1) \\
&= P(x_n | x_{n-1}, \ldots x_1) P(x_{n-1} | x_{n-2} \ldots x_1) \ldots P(x_2 | x_1) P(x_1) \\
&= \Pi_{i=1} \mathbf{P}(x_i | x_{i-1}, \ldots x_1) \\
&= \Pi_{i=1} \mathbf{P}(x_i | Parents(X_i))
\end{aligned}$$

- What is the distribution at a single node, given the rest of the network and the evidence $\mathbf{e}$?
- **Parents** of $\mathbf{X}$, the set $\mathcal{P}$ are the nodes on which $\mathbf{X}$ is conditioned.
- **Children** of $\mathbf{X}$, the set $\mathcal{C}$ are the nodes conditioned on $\mathbf{X}$.

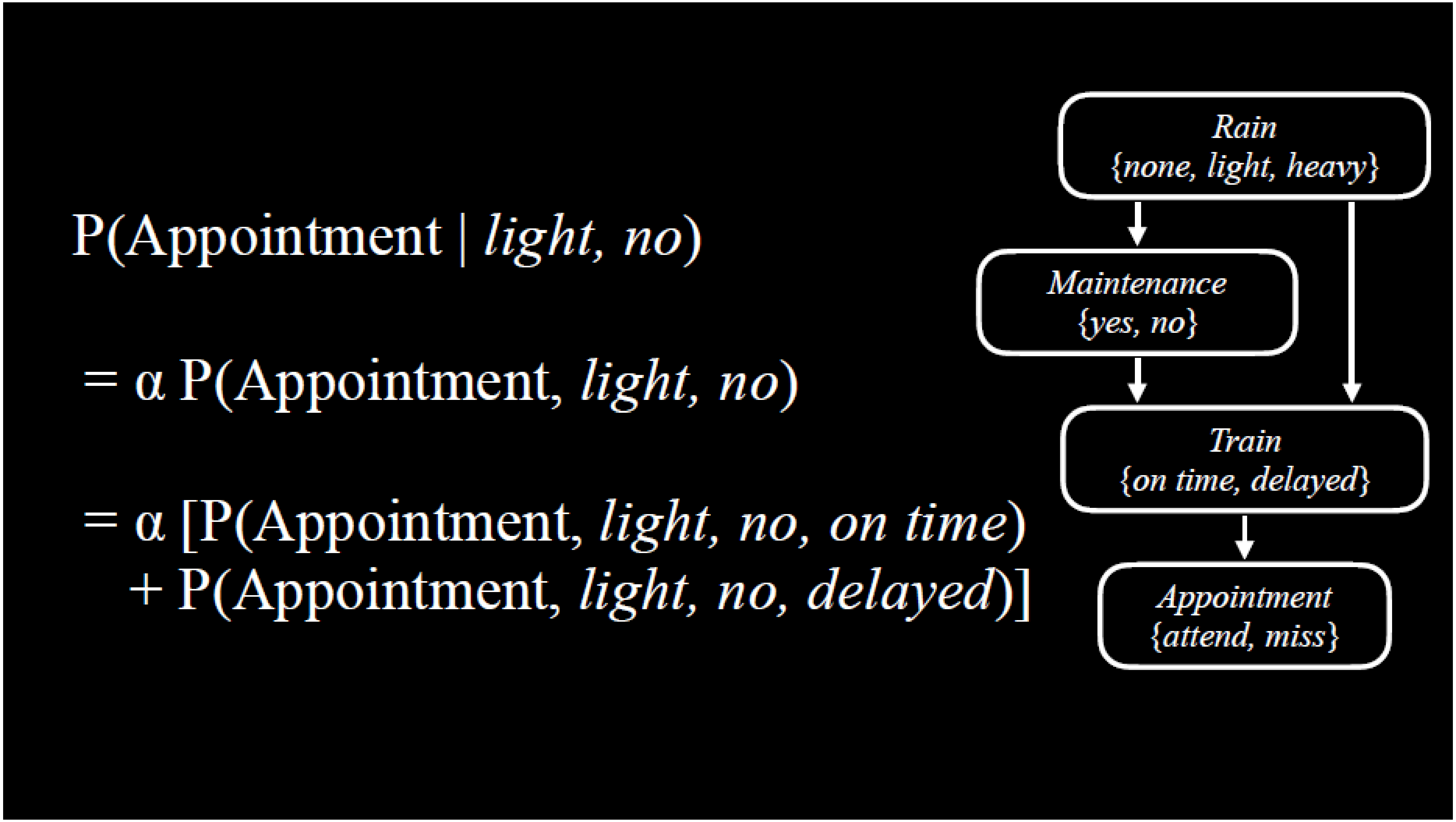


- Use the Bayes Rule, for the case on the right:

$$\begin{aligned} P(a,b,x,c,d) &= P(a,b,x|c,d)P(c,d) \\ &= P(a,b|x)P(x|c,d)P(c,d) \end{aligned}$$

or more generally,

$$P(\mathcal{C}(x),x,\mathcal{P}(x)|\mathbf{e}) = P(\mathcal{C}(x)|x,\mathbf{e})P(x|\mathcal{P}(x),\mathbf{e})P(\mathcal{P}(x)|,\mathbf{e})$$



$$P(\text{Appointment} \mid \textit{light}, \textit{no})$$

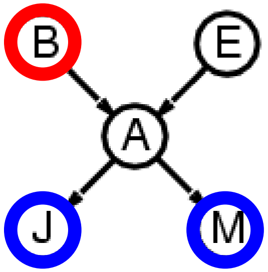
$$= \alpha\, P(\text{Appointment}, \textit{light}, \textit{no})$$

$$= \alpha\, [P(\text{Appointment}, \textit{light}, \textit{no}, \textit{on time}) + P(\text{Appointment}, \textit{light}, \textit{no}, \textit{delayed})]$$

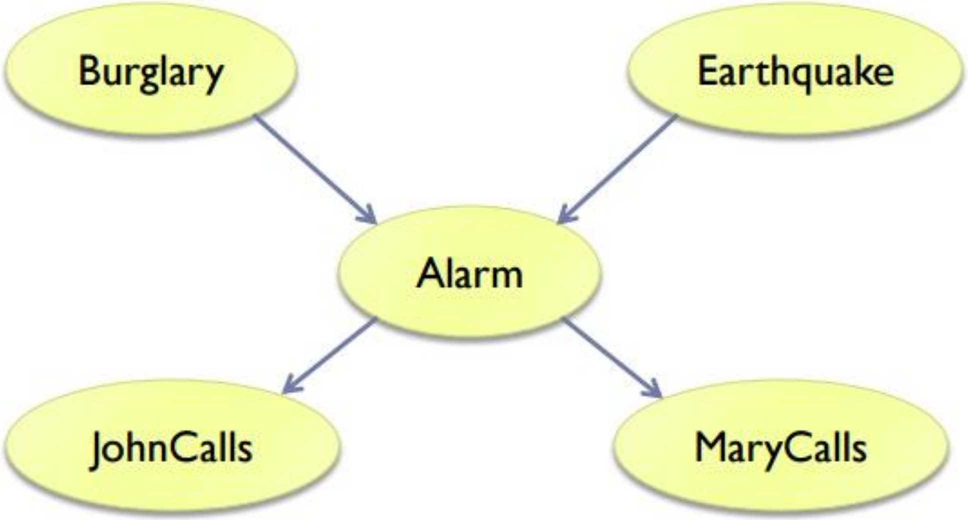
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Burglary example

- Query: P(b | j, m)



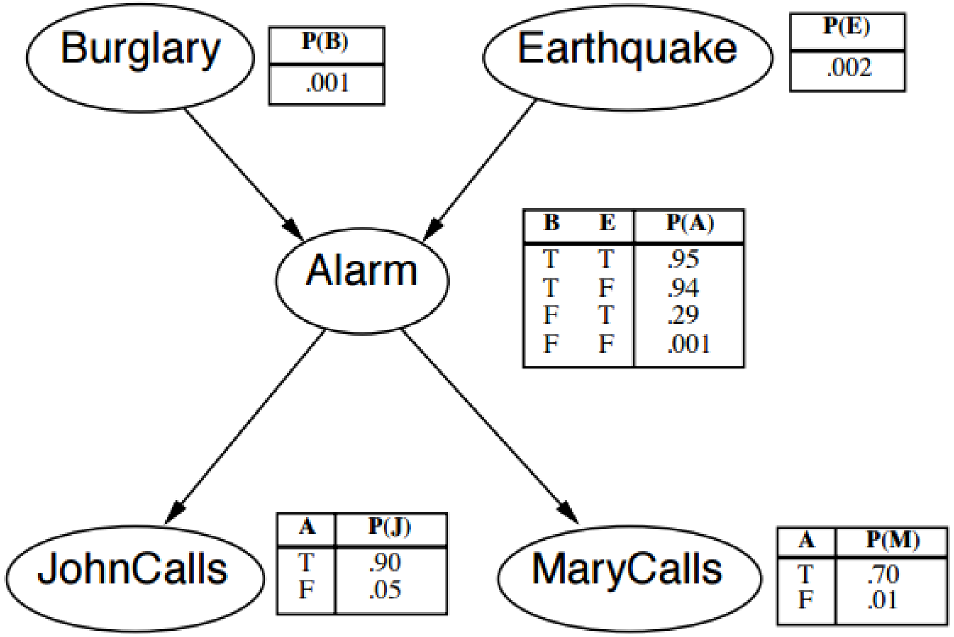
$$P(b \mid j, m)$$



$$\boldsymbol{P}\,(x_1,\ \ldots\ ,x_n) = \prod_{i=1} \boldsymbol{P}\,(x_i \mid Parents(X_i))$$

P(JohnCalls ^ MaryCalls ^ Alarm ^ Burglary ^ Earthquake)
= P(JohnCalls|Alarm) x P(MaryCalls|Alarm) x P(Alarm|Burglary^Earthquake)
x P(Burglary) x P(Earthquake)

Example



P(B)
.001

P(E)
.002

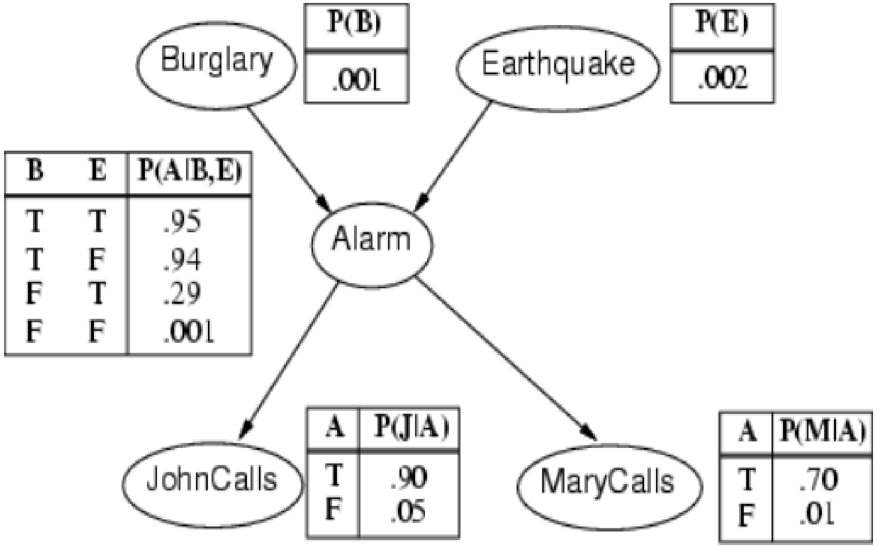
B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

A	P(M)
T	.70
F	.01

- Key: given knowledge of the values of some nodes in the network, we can apply Bayesian inference to determine the maximum posterior values of the unknown variables!

Problem 1



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

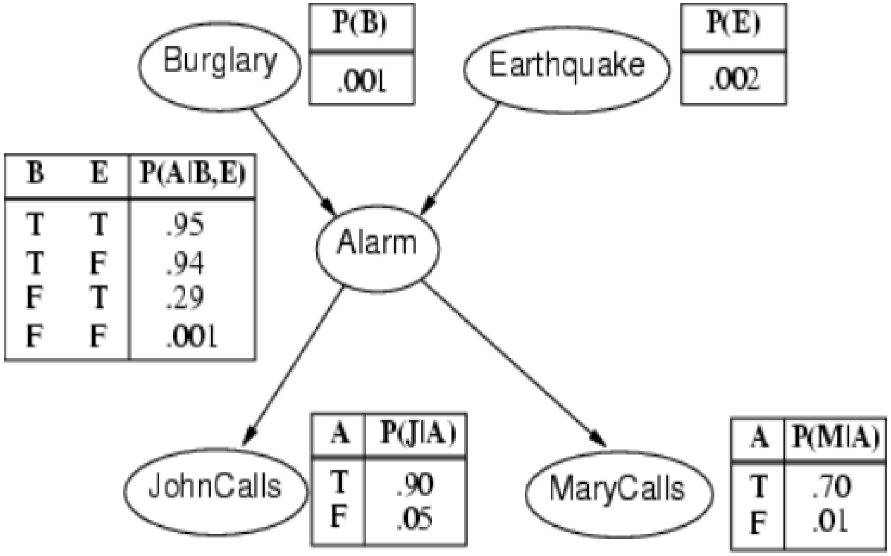
A	P(M|A)
T	.70
F	.01

J: JohnCalls
M: MaryCalls
A: Alarm
B: Burglary
E: Earthquake

What is the probability of the event that the alarm has sounded and no burglary but an earthquake has occurred and both Mary and John call?

$$P(J \wedge M \wedge A \wedge \sim B \wedge E) = P(J|A) \times P(M|A) \times P(A|\sim B \wedge E) \times P(\sim B) \times P(E)$$
$$= 0.90 \times 0.70 \times 0.29 \times 0.999 \times 0.002 = 0.00036$$

Problem 2



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

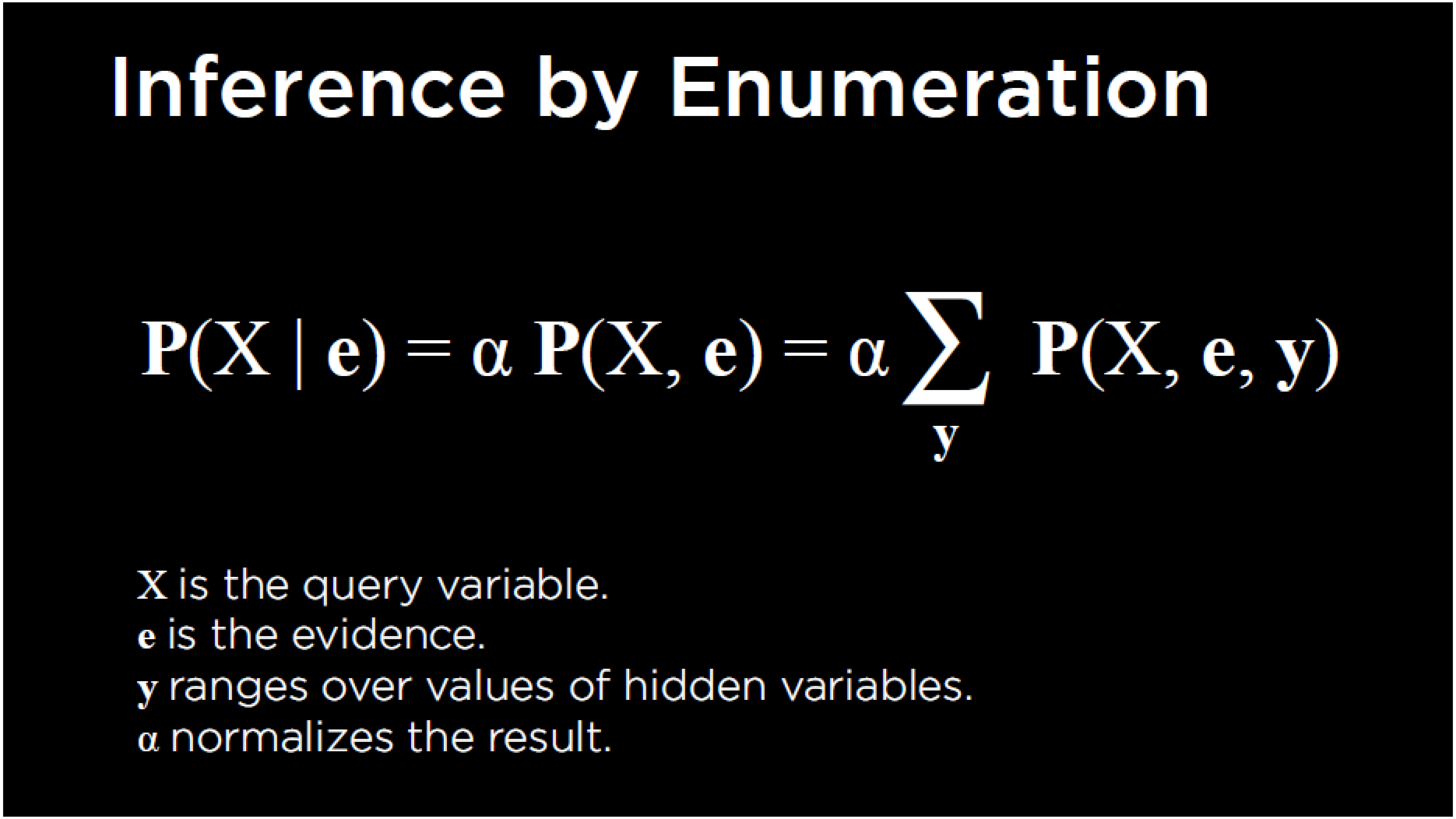
A	P(M|A)
T	.70
F	.01

J: JohnCalls
M: MaryCalls
A: Alarm
B: Burglary
E: Earthquake

What is the probability of the event that the alarm has sounded but neither a burglary nor an earthquake has occurred and John call and Mary didn't call?

$$P(J \wedge \sim M \wedge A \wedge \sim B \wedge \sim E) = P(J|A) \times P(\sim M|A) \times P(A|\sim B \wedge \sim E) \times P(\sim B) \times P(\sim E)$$

$$= 0.90 \times 0.30 \times 0.001 \times 0.999 \times 0.998 = 0.00027$$



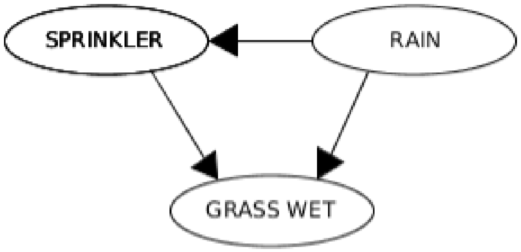
Inference by Enumeration

$$\mathbf{P}(X \mid \mathbf{e}) = \alpha\, \mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

X is the query variable.
$\mathbf{e}$ is the evidence.
$\mathbf{y}$ ranges over values of hidden variables.
α normalizes the result.

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

What is P(S|G)?

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

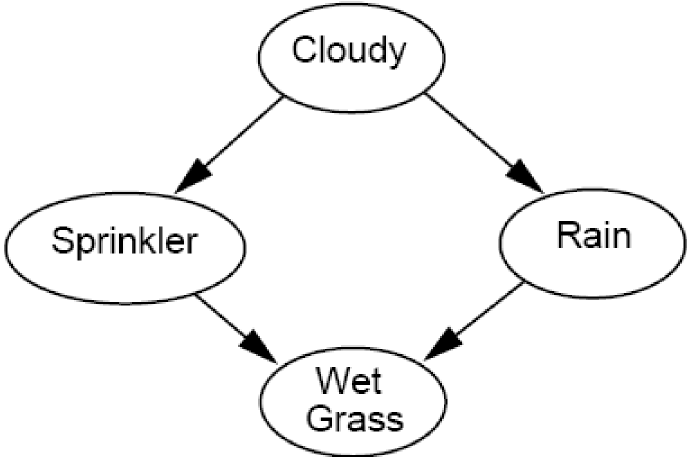
P(S|G) = P(S^G)/P(G)
0.6467

P(S^G) = P(S^G^R) + P(S^G^~R) = 0.00198+0.288 = .28998
P(G) = P(S^G) + P(~S^G) = .28998 + 0.1584 = 0.44838

P(~S^G) =P(~S^G^R) + P(~S^G^~R) = 0.1584 + 0
P(S^G^R)=P(S|R)P(G|S^R)P(R) = (0.01)(0.99)(0.2) = 0.00198
P(S^G^~R) = P(S|~R)P(G|S^~R)P(~R) = (0.4)(0.9)(0.8) = 0.288
P(~S^G^R) = P(~S|R)P(G|~S^R)P(R) = (0.99)(0.8)(0.2) = 0.1584
P(~S^G^~R) = P(~S|~R)P(G|~S^~R)P(~R) = (0.6)(0.0)(0.8) = 0

Another example

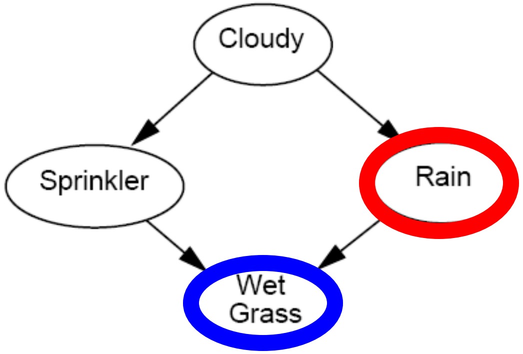
- Variables: *Cloudy, Sprinkler, Rain, WetGrass*



Another example

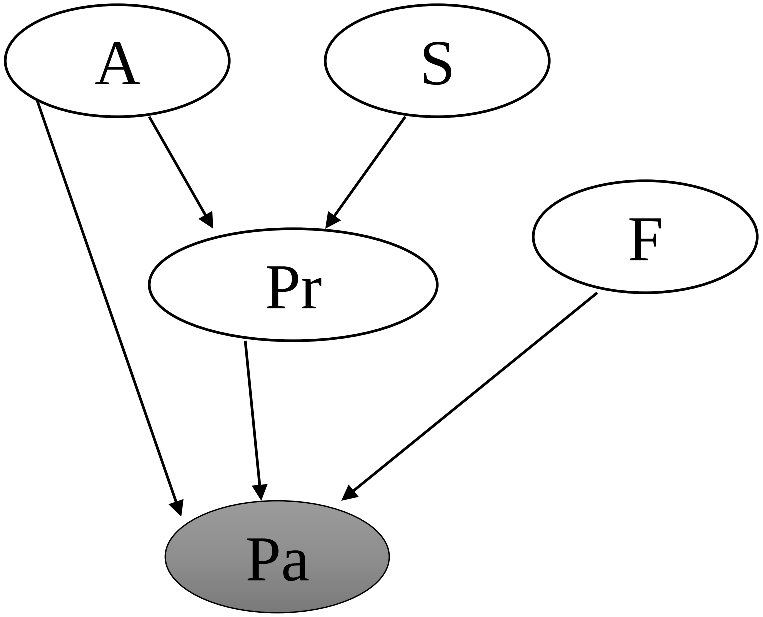
- Given that the grass is wet, what is the probability that it has rained?

$$P(r \mid w)$$

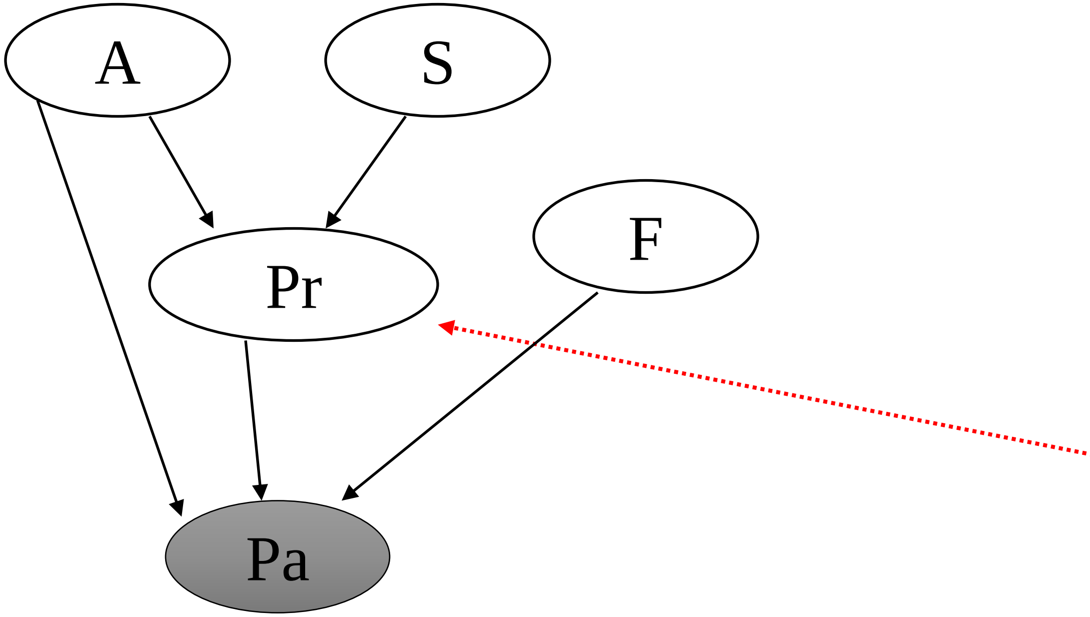


Another example

- What determines whether you will pass the exam?
  - **A**: Do you attend class?
  - **S**: Do you study?
  - **Pr**: Are you prepared for the exam?
  - **F**: Is the grading fair?
  - **Pa**: Do you get a passing grade on the exam?



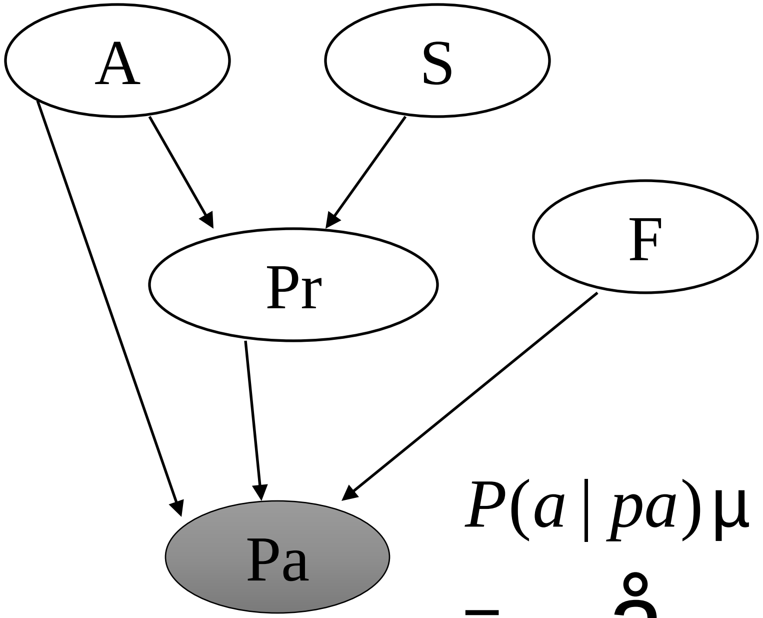
Another example



A	S	P(Pr|A,S)
T	T	0.9
T	F	0.5
F	T	0.7
F	F	0.1

Pr	A	F	P(Pa|A,Pr,F)
T	T	T	0.9
T	T	F	0.6
T	F	T	0.2
T	F	F	0.1
F	T	T	0.4
F	T	F	0.2
F	F	T	0.1
F	F	F	0.2

Another example

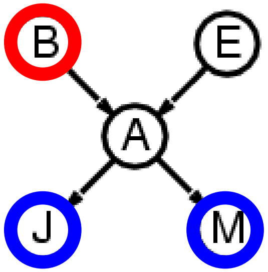


$$P(a \mid pa) \propto P(a, pa)$$

$$= \sum_{S=s, F=f, Pr=pr} P(a, s, f, pr, pa)$$

$$= \sum_{S=s, F=f, Pr=pr} P(a)P(s)P(f)P(pr \mid a, s)P(pa \mid a, pr, f)$$

Efficient inference



- Query: P(b | j, m)

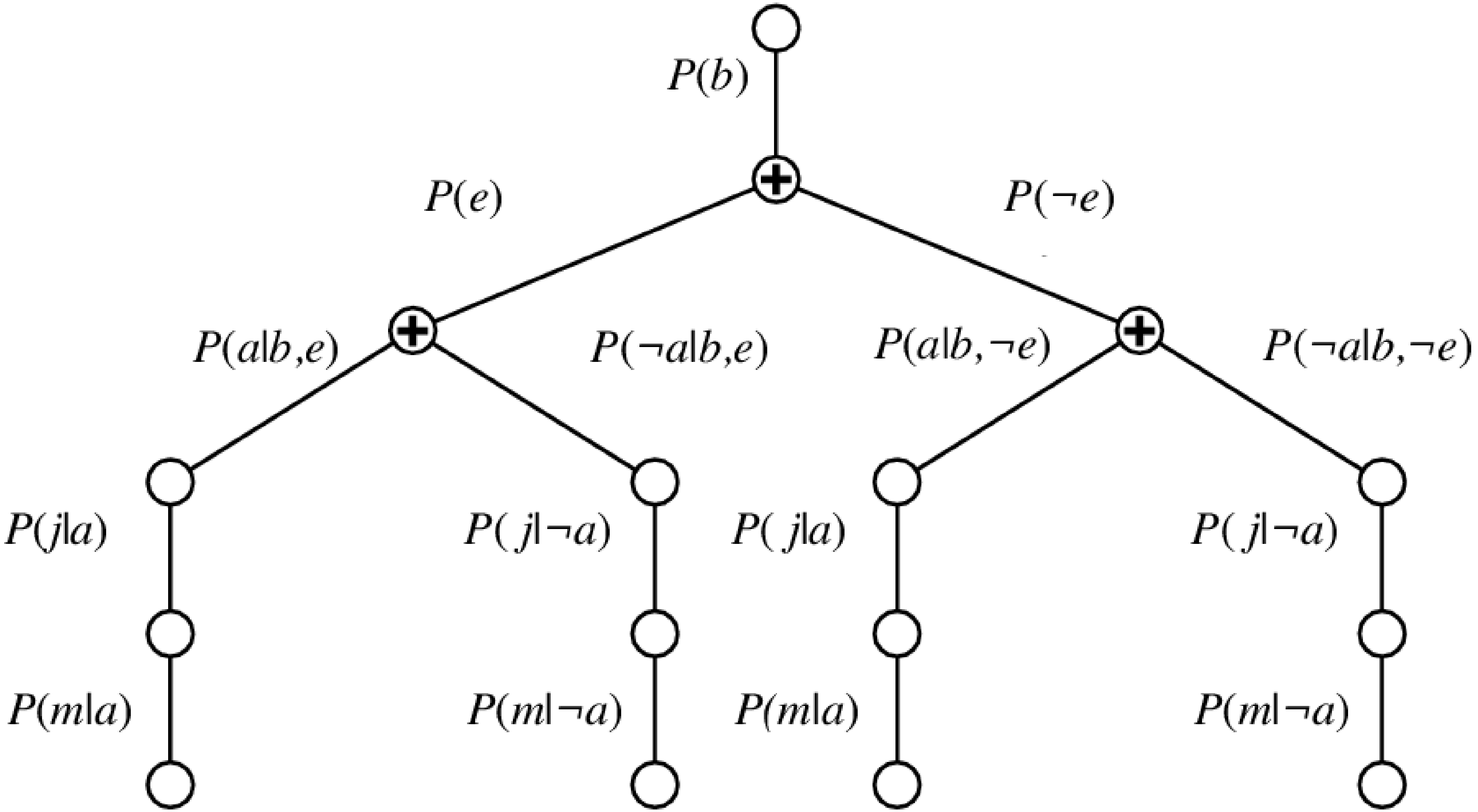
$$P(b \mid j, m) = \frac{P(b, j, m)}{P(j, m)} \mu \underset{E=e, A=a}{\mathring{a}} P(b, e, a, j, m)$$

$$= \underset{E=e, A=a}{\mathring{a}} P(b) P(e) P(a \mid b, e) P(j \mid a) P(m \mid a)$$

- Can we compute this sum efficiently?

Efficient inference

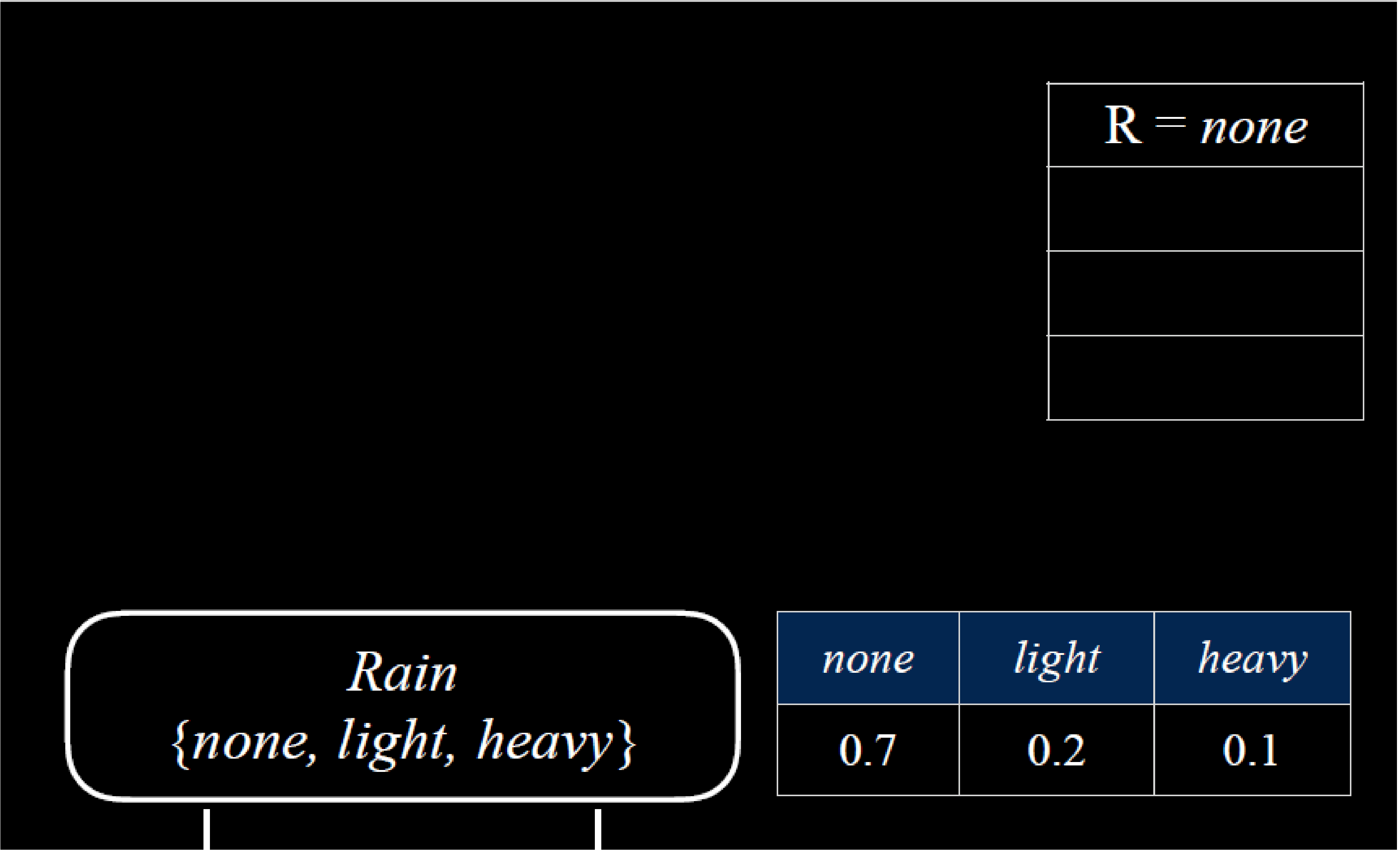
$$P(b \mid j, m) \propto P(b) \sum_{E=e} P(e) \sum_{A=a} P(a \mid b, e) P(j \mid a) P(m \mid a)$$



Efficient inference

- Key idea: compute the results of sub-expressions in a bottom-up way and cache them for later use
  - Form of dynamic programming
  - Polynomial time and space complexity for *polytrees*: networks at most one undirected path between any two nodes

Approximate Inference : Sampling



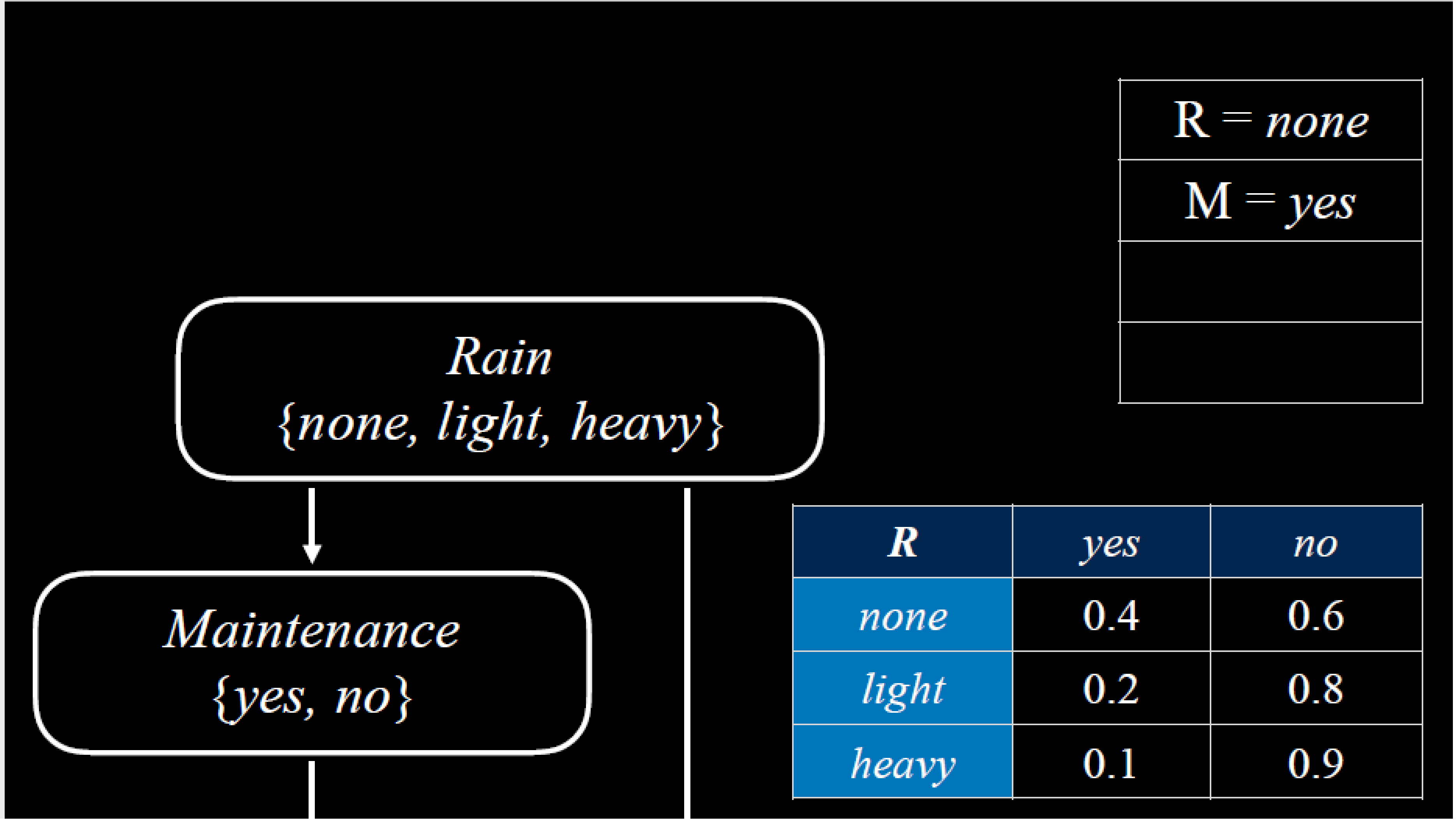
R = *none*

Rain
{none, light, heavy}

none	*light*	*heavy*
0.7	0.2	0.1

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sampling



Rain
{none, light, heavy}

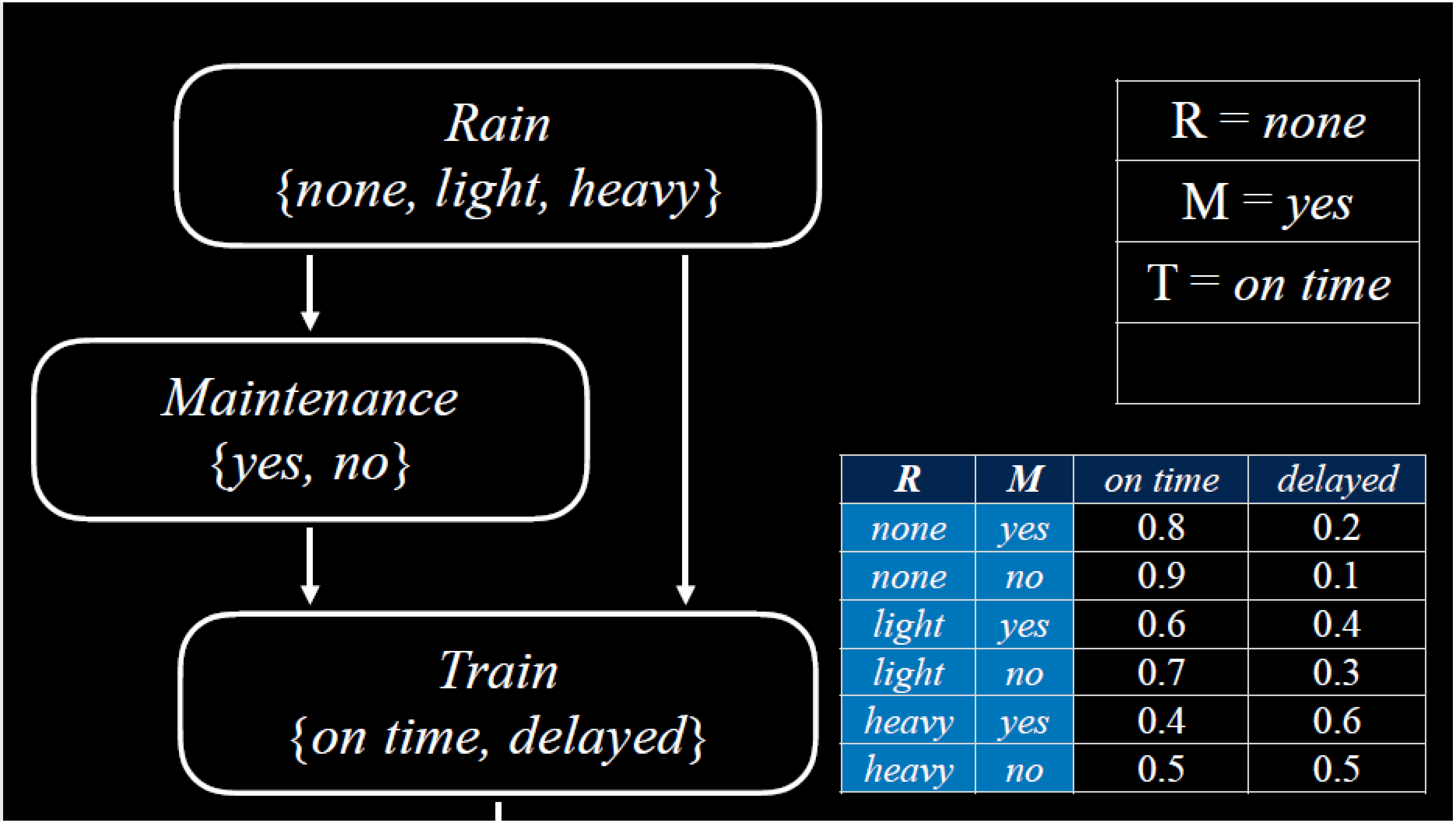
Maintenance
{yes, no}

R = *none*
M = *yes*

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sampling

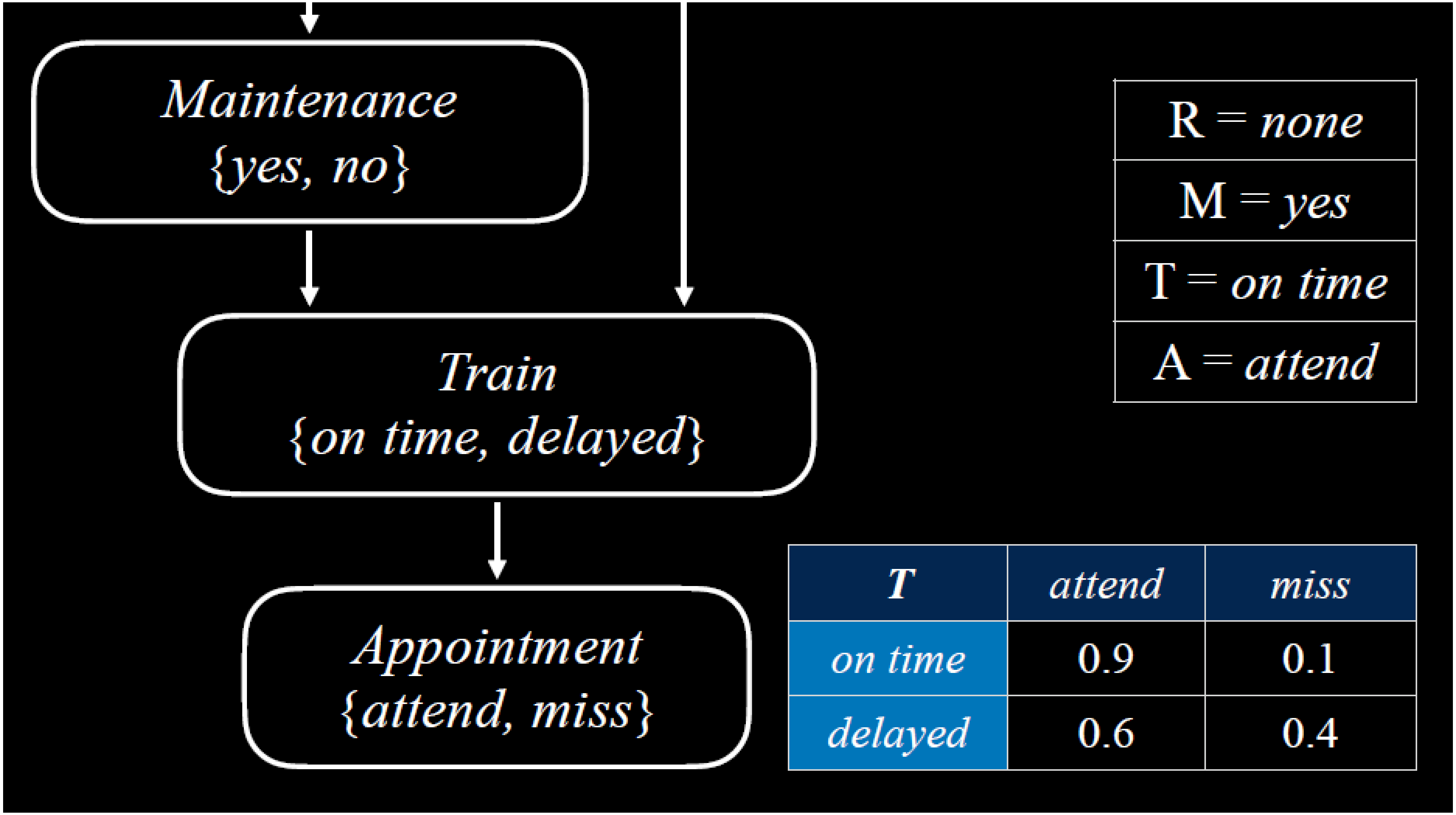


R = *none*
M = *yes*
T = *on time*

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sampling

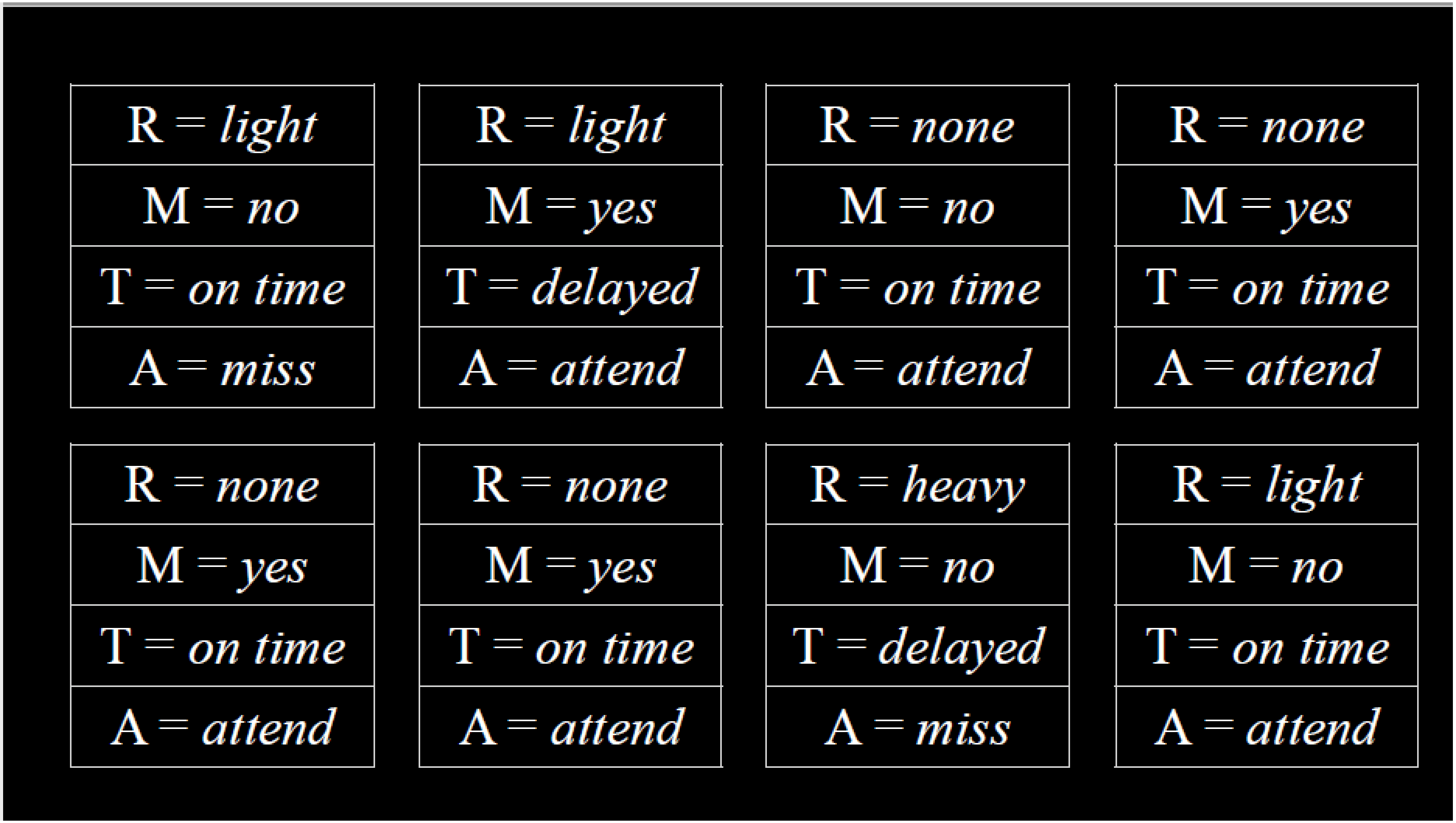


R = *none*
M = *yes*
T = *on time*
A = *attend*

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sampling



R = *light*
M = *no*
T = *on time*
A = *miss*

R = *light*
M = *yes*
T = *delayed*
A = *attend*

R = *none*
M = *no*
T = *on time*
A = *attend*

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

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

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

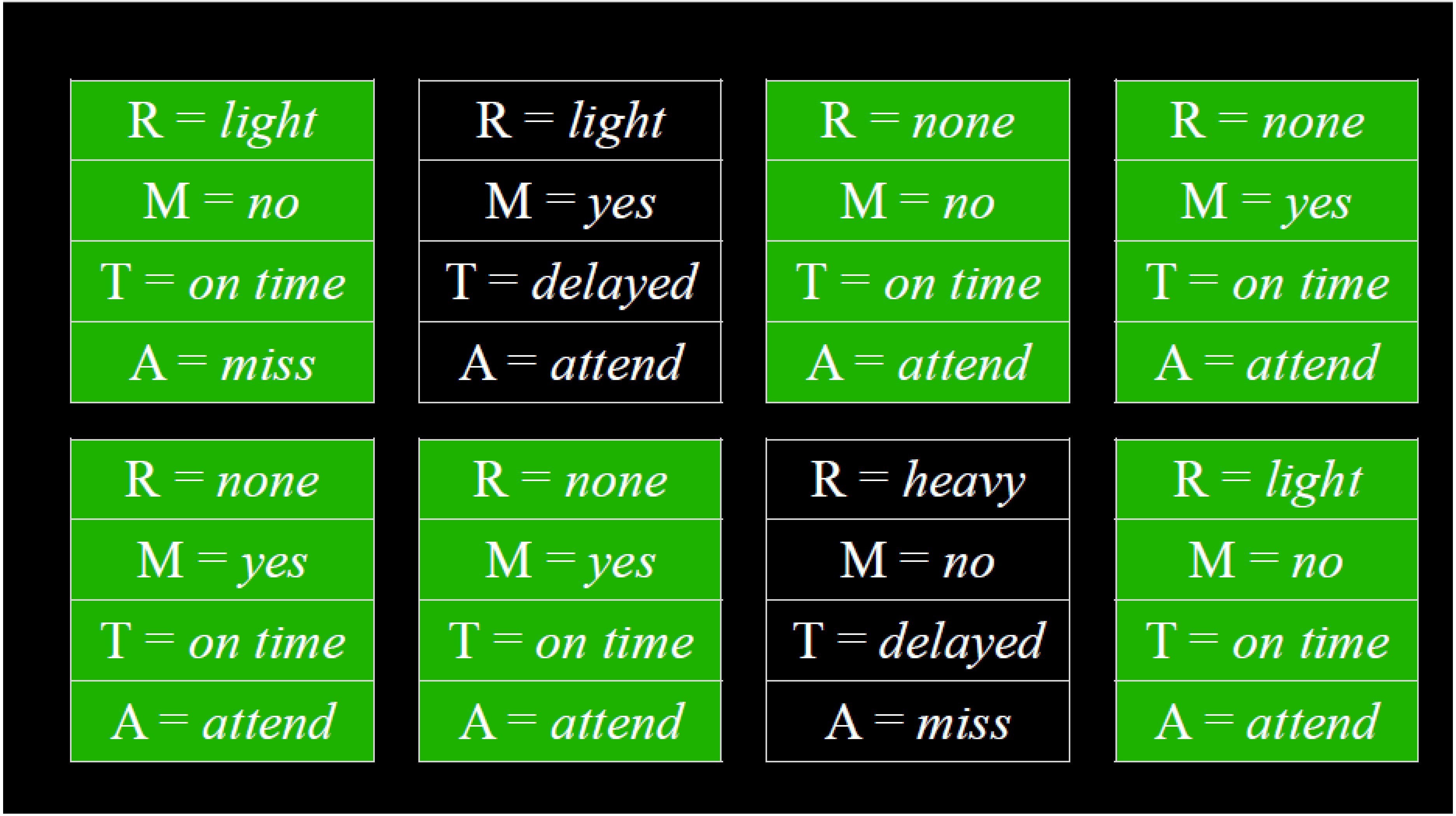
R = *heavy*
M = *no*
T = *delayed*
A = *miss*

R = *light*
M = *no*
T = *on time*
A = *attend*

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sampling

$$P(Train = on\ time)$$

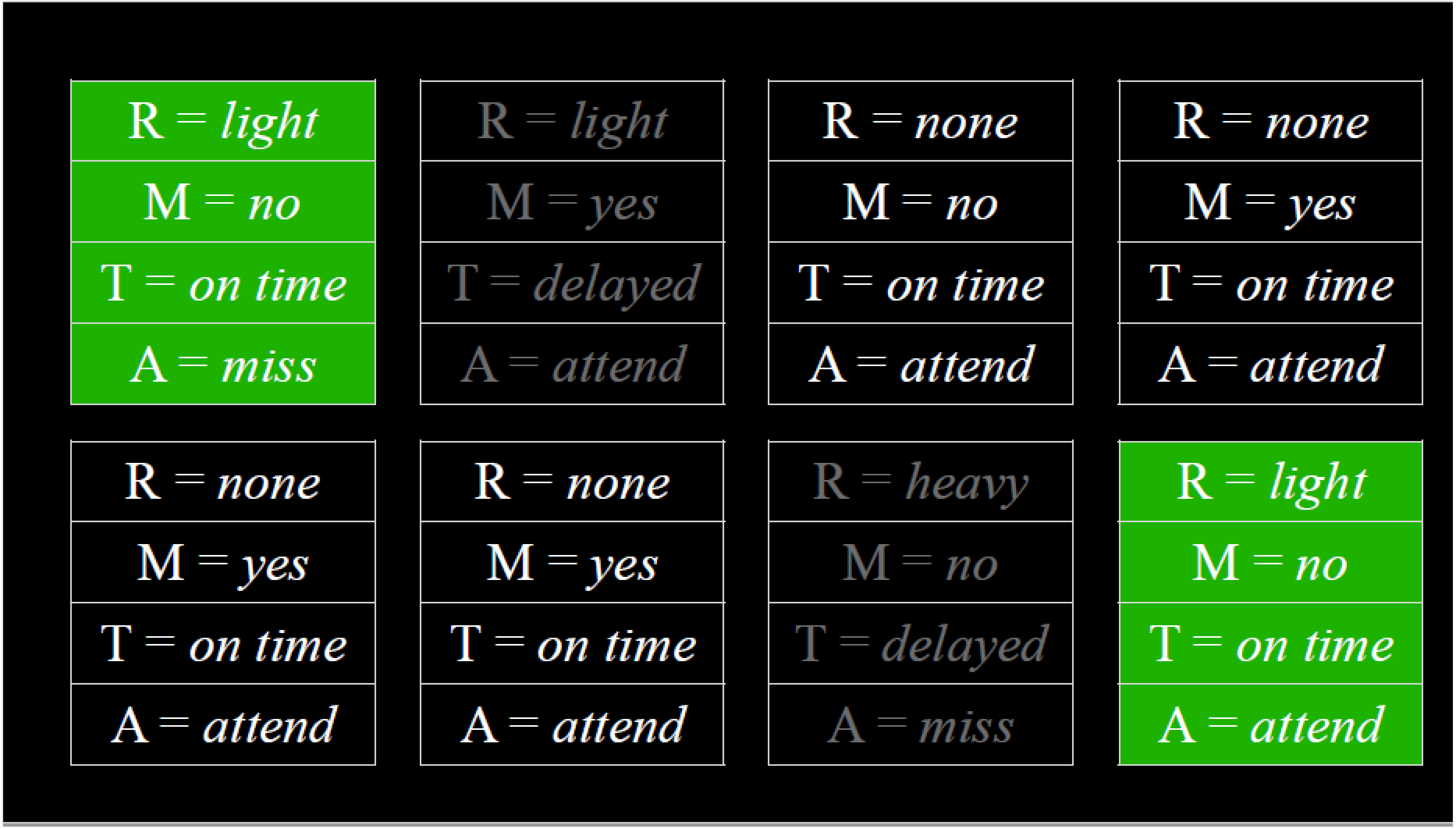


🟩 $R = light$ $M = no$ $T = on\ time$ $A = miss$	$R = light$ $M = yes$ $T = delayed$ $A = attend$	🟩 $R = none$ $M = no$ $T = on\ time$ $A = attend$	🟩 $R = none$ $M = yes$ $T = on\ time$ $A = attend$
🟩 $R = none$ $M = yes$ $T = on\ time$ $A = attend$	🟩 $R = none$ $M = yes$ $T = on\ time$ $A = attend$	$R = heavy$ $M = no$ $T = delayed$ $A = miss$	🟩 $R = light$ $M = no$ $T = on\ time$ $A = attend$

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Sa

P(Rain = *light* | Train = *on time*)



R = *light*
M = *no*
T = *on time*
A = *miss*

R = *light*
M = *yes*
T = *delayed*
A = *attend*

R = *none*
M = *no*
T = *on time*
A = *attend*

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

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

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

R = *heavy*
M = *no*
T = *delayed*
A = *miss*

R = *light*
M = *no*
T = *on time*
A = *attend*

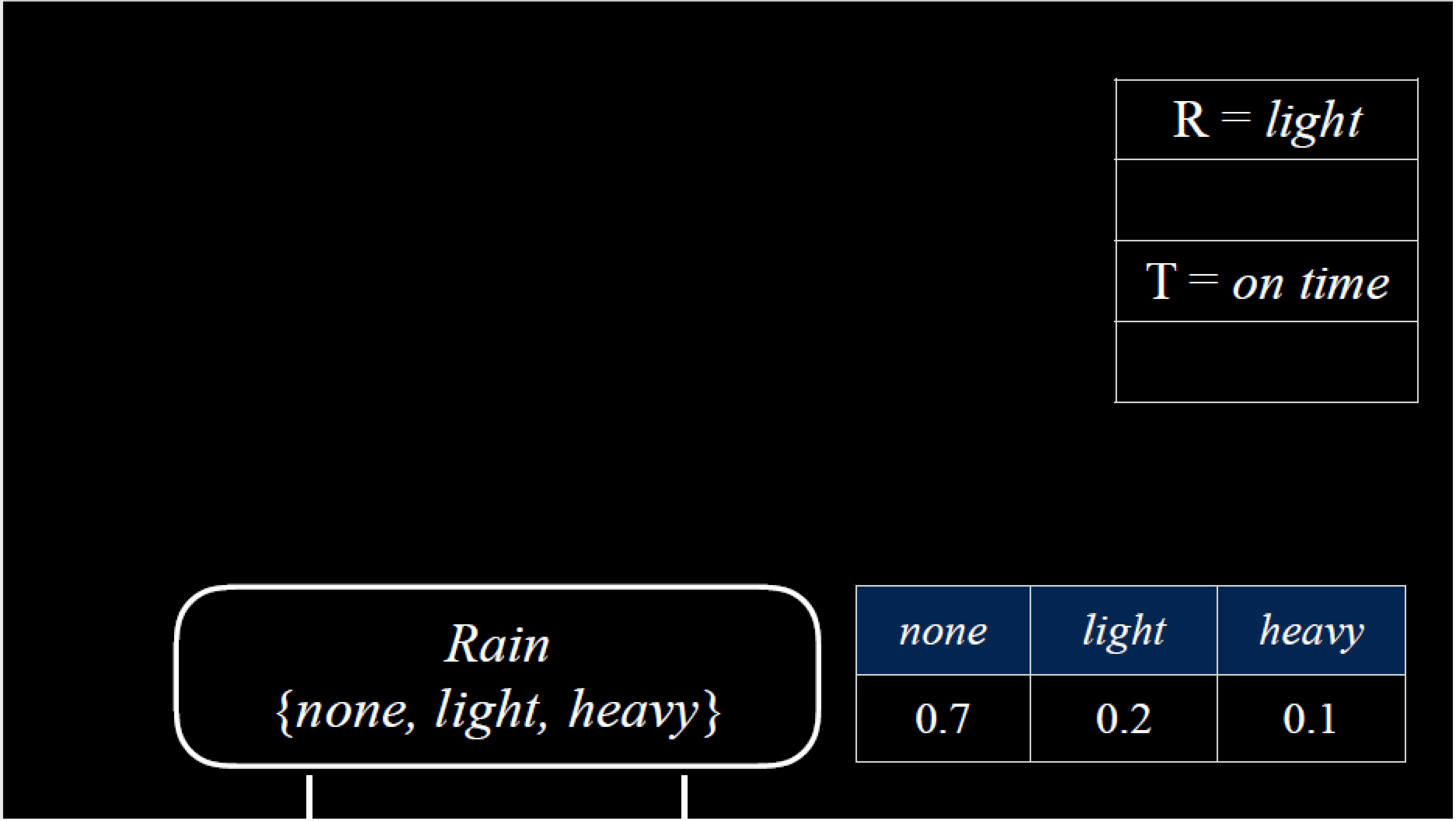
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – 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 of the evidence.

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – Likelihood Weighting



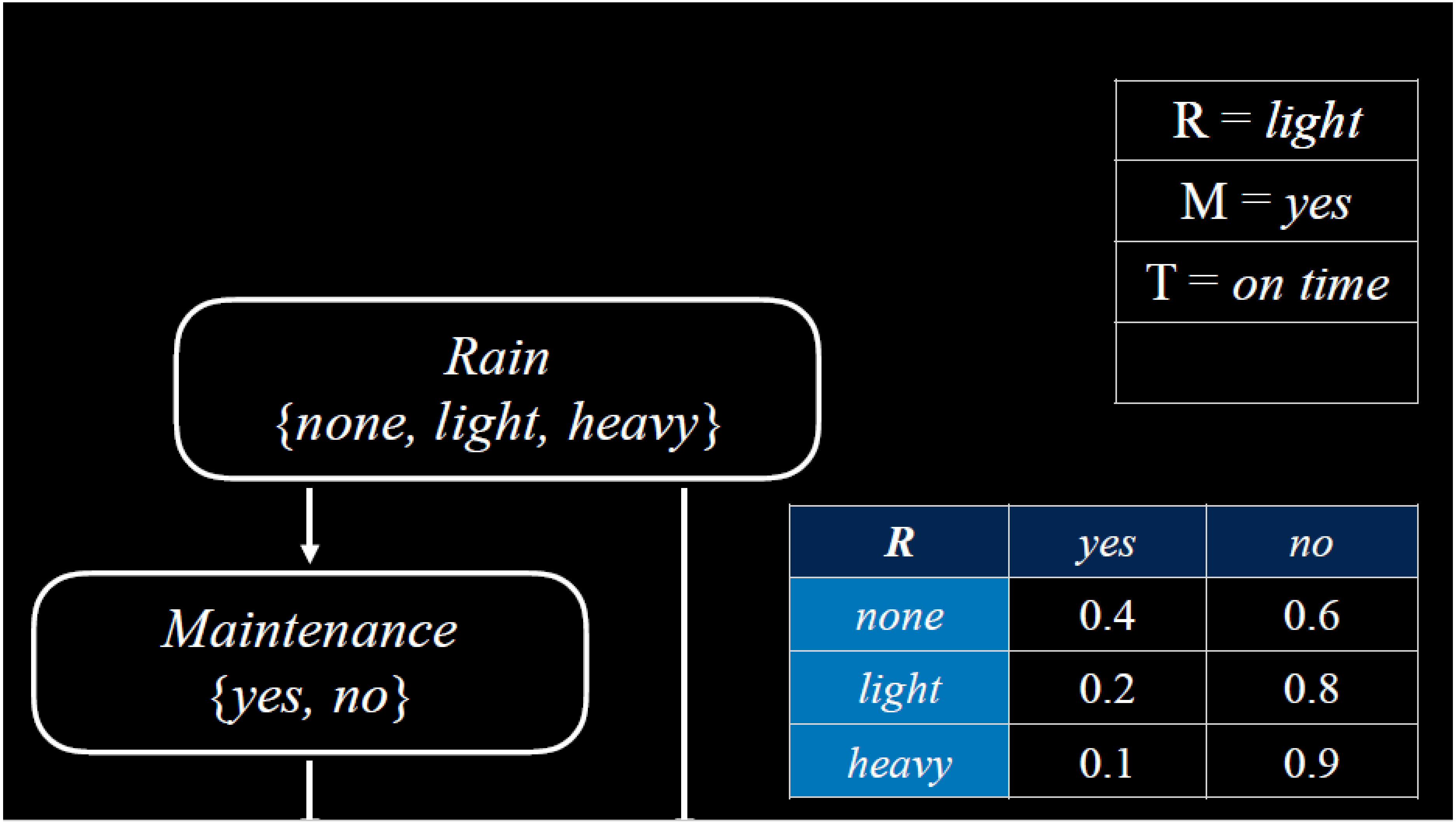
R = *light*
T = *on time*

Rain
{none, light, heavy}

none	*light*	*heavy*
0.7	0.2	0.1

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – Likelihood Weighting



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

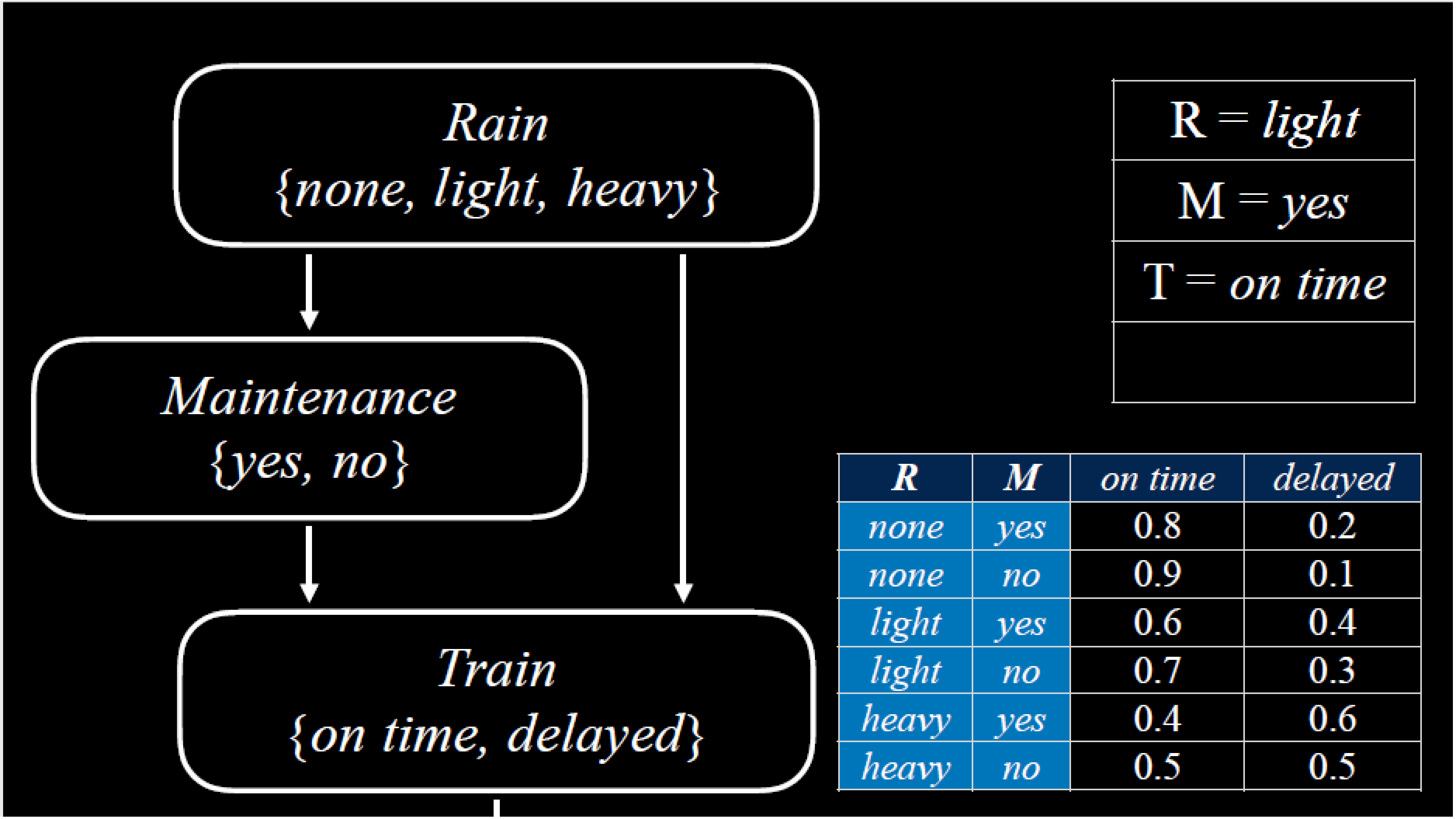
Maintenance
{*yes*, *no*}

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – Likelihood Weighting



Rain
{none, light, heavy}

Maintenance
{yes, no}

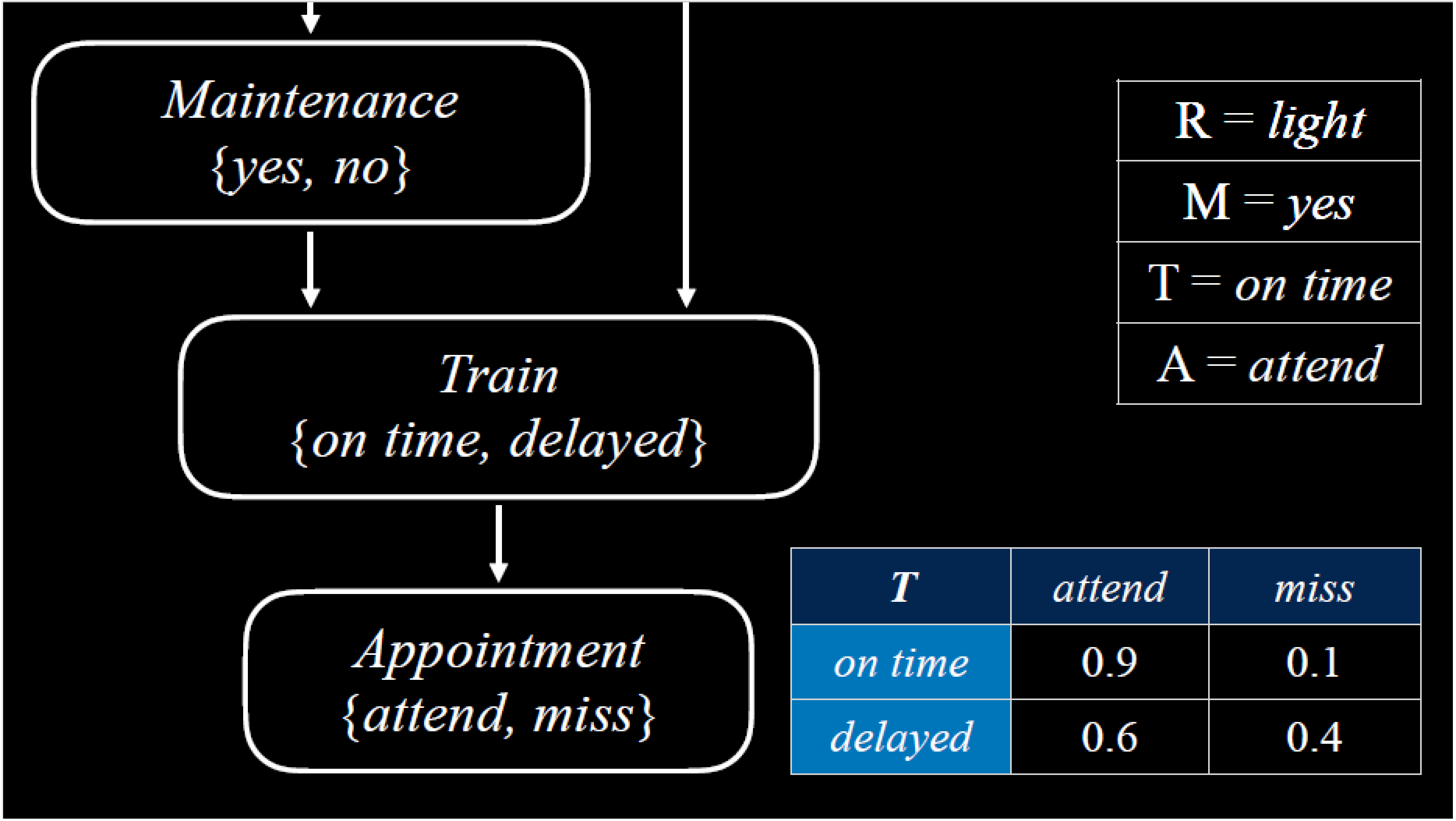
Train
{on time, delayed}

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – Likelihood Weighting



Maintenance
{*yes, no*}

Train
{*on time, delayed*}

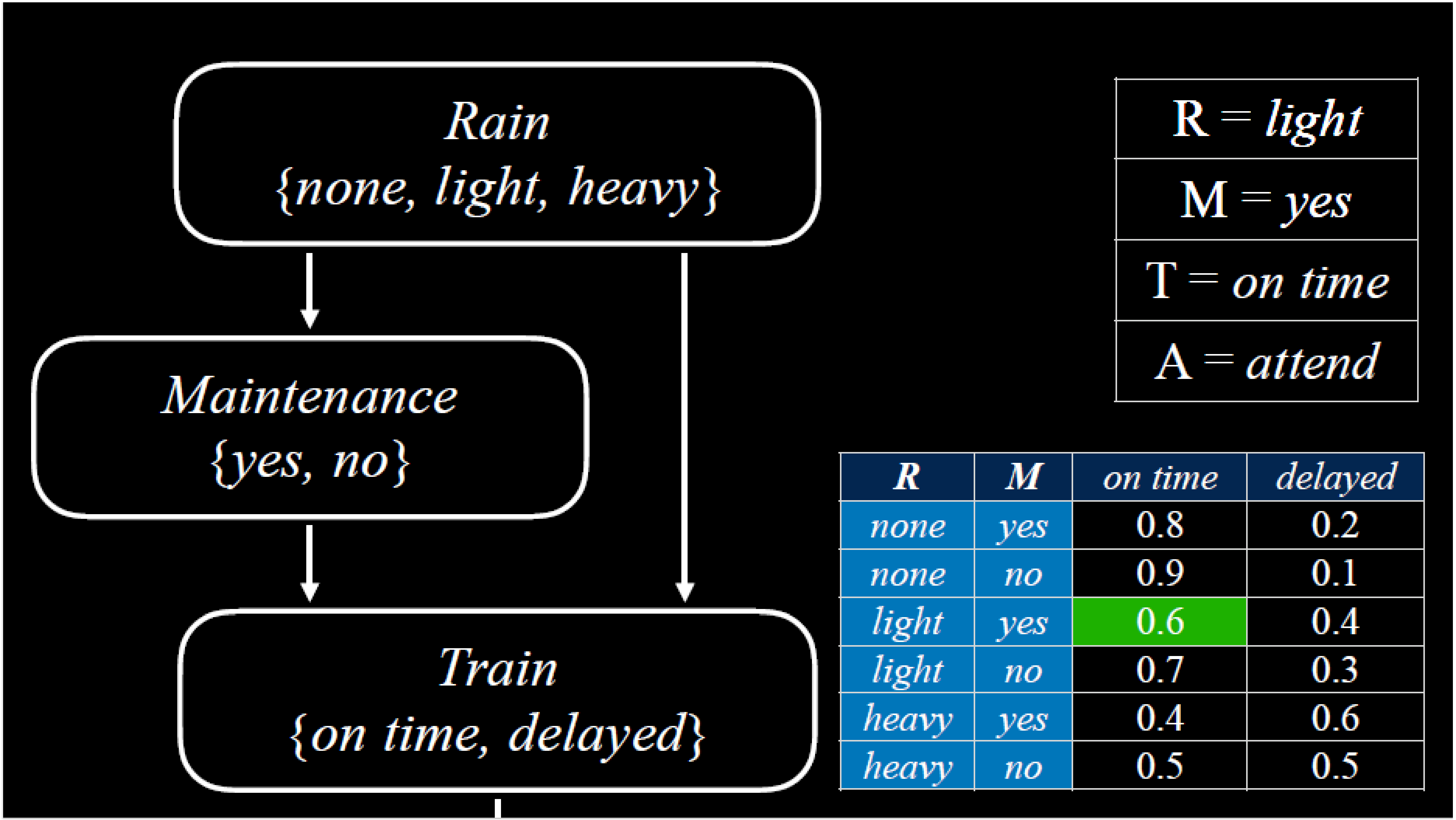
Appointment
{*attend, miss*}

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

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

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Approximate Inference : Rejection Sampling – Likelihood Weighting



R = *light*

M = *yes*

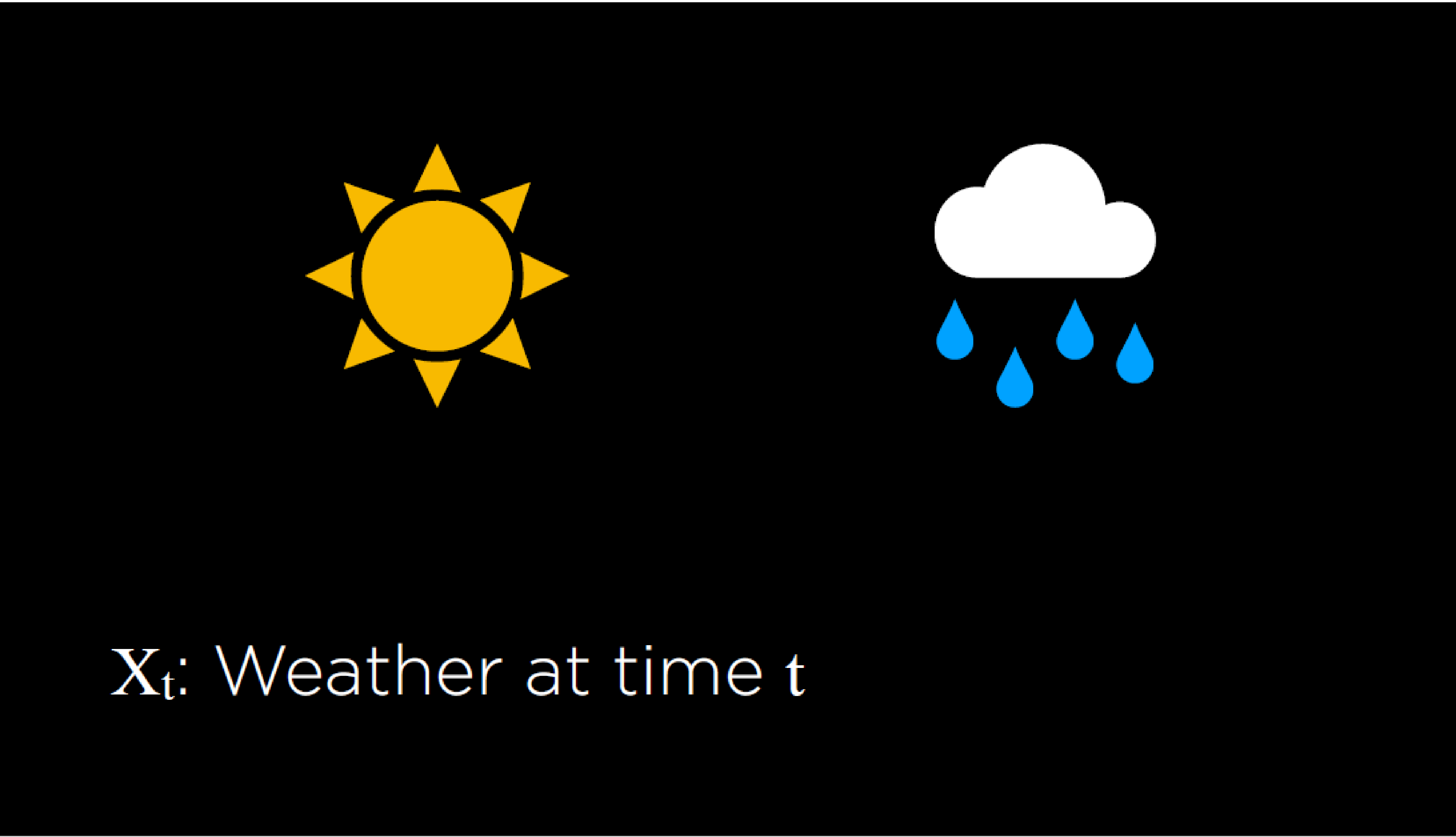
T = *on time*

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

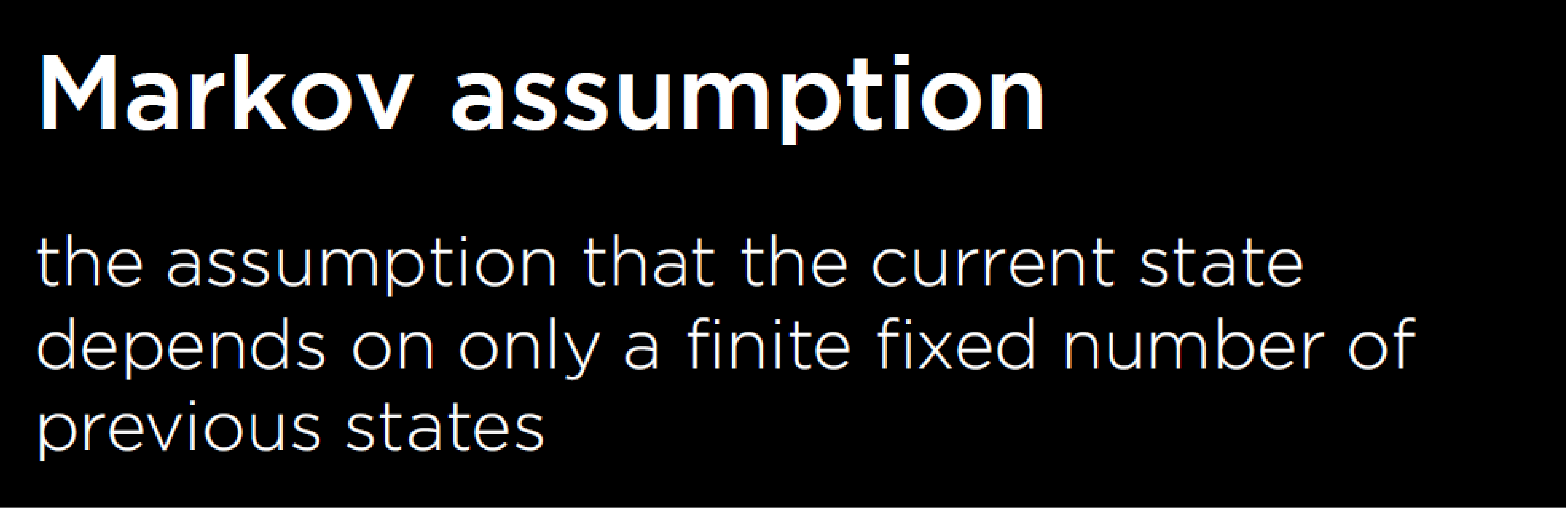
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time



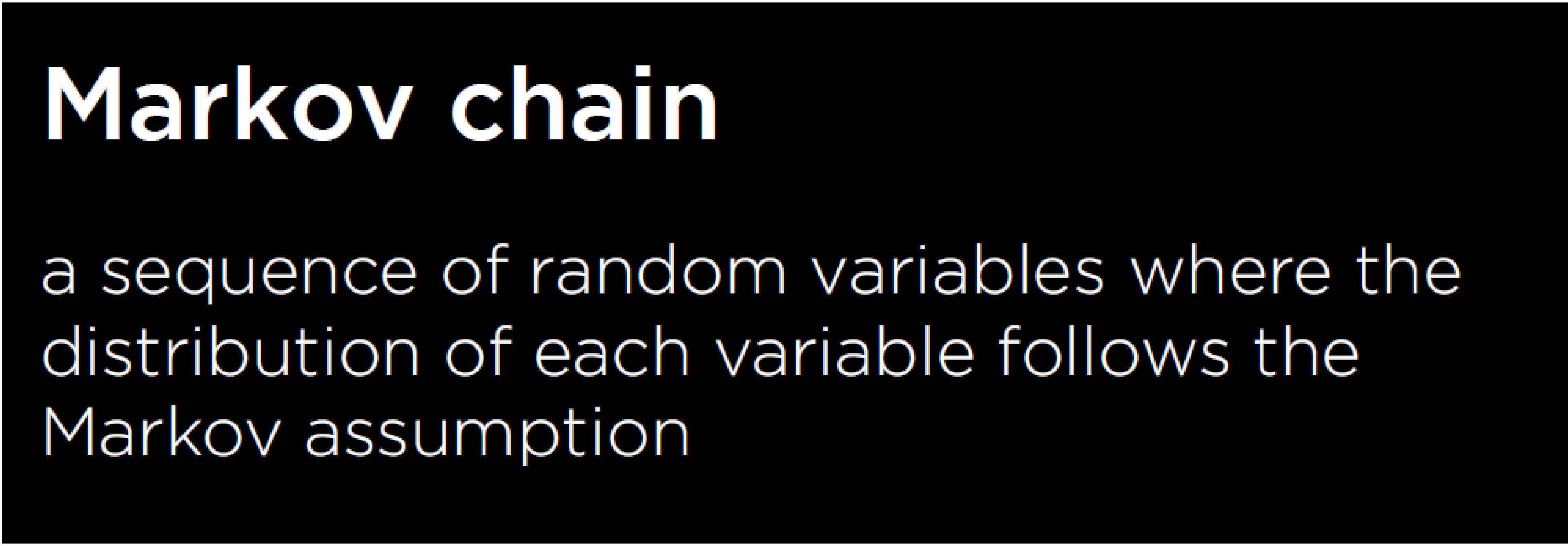
X_t: Weather at time t

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu



Markov assumption

the assumption that the current state depends on only a finite fixed number of previous states

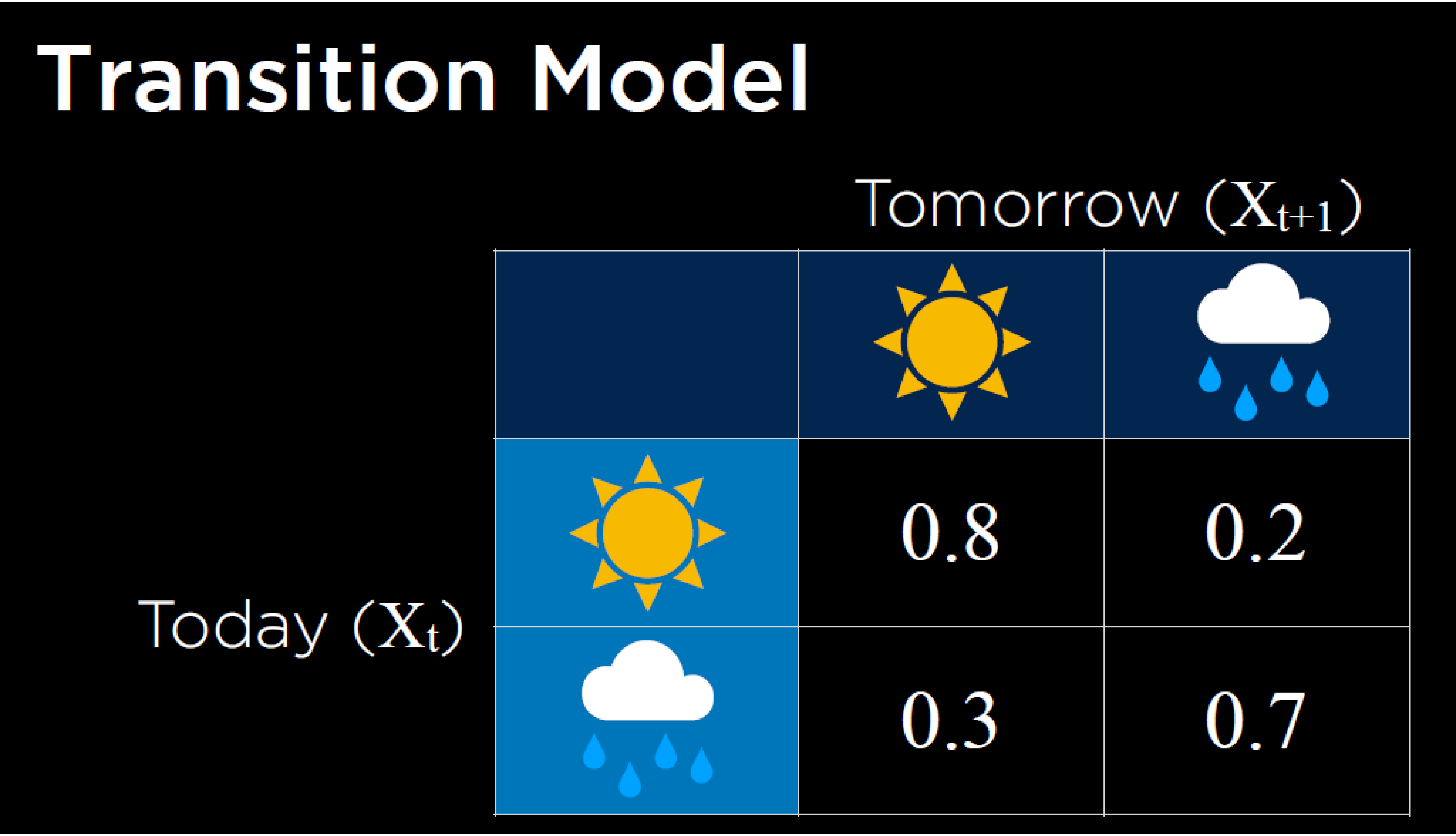


Markov chain

a sequence of random variables where the distribution of each variable follows the Markov assumption

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

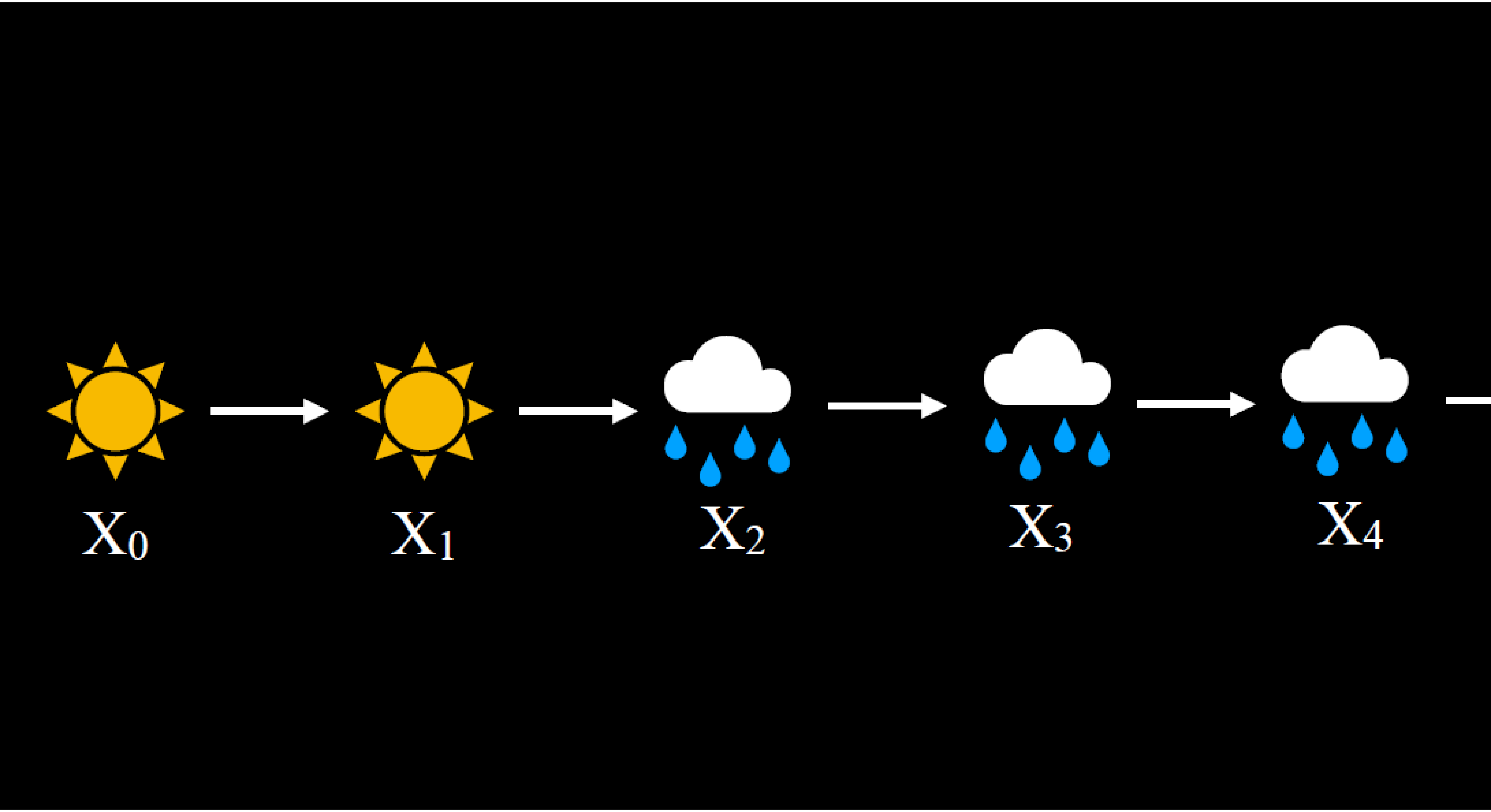


Transition Model

		Tomorrow (X_{t+1})	
		☀	🌧
Today (X_t)	☀	0.8	0.2
	🌧	0.3	0.7

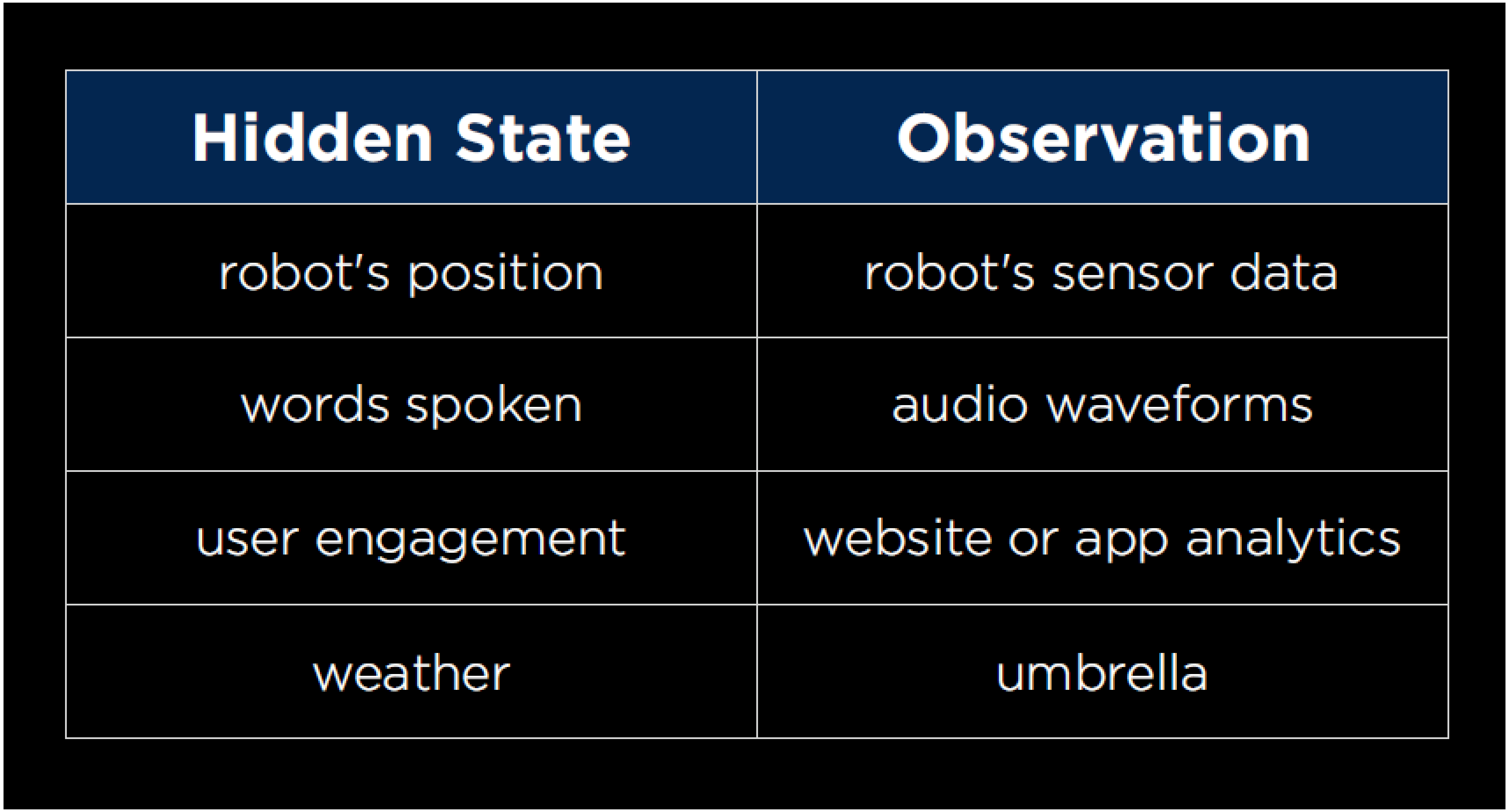
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time



Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

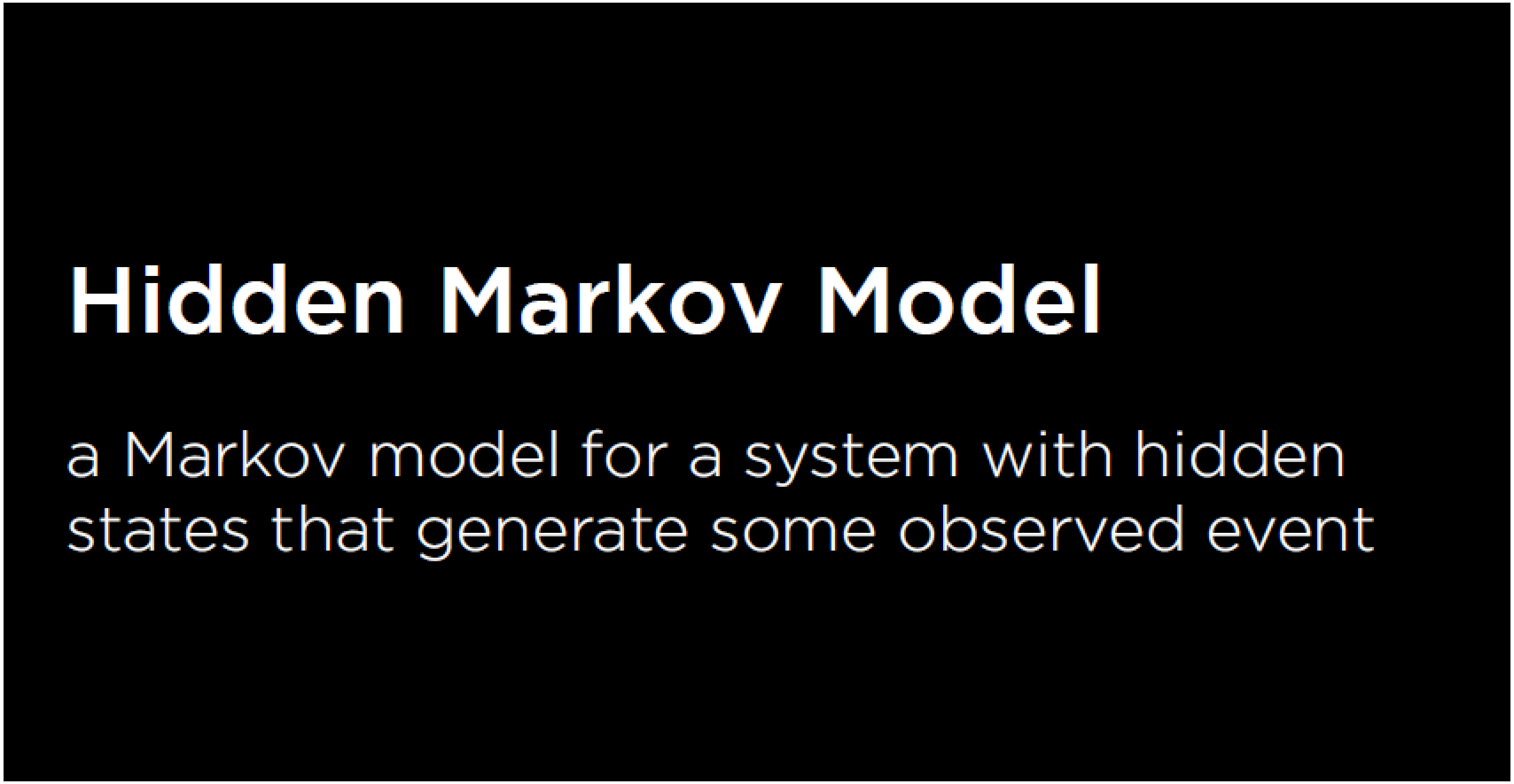
Sensor Models



Hidden State	Observation
robot's position	robot's sensor data
words spoken	audio waveforms
user engagement	website or app analytics
weather	umbrella

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

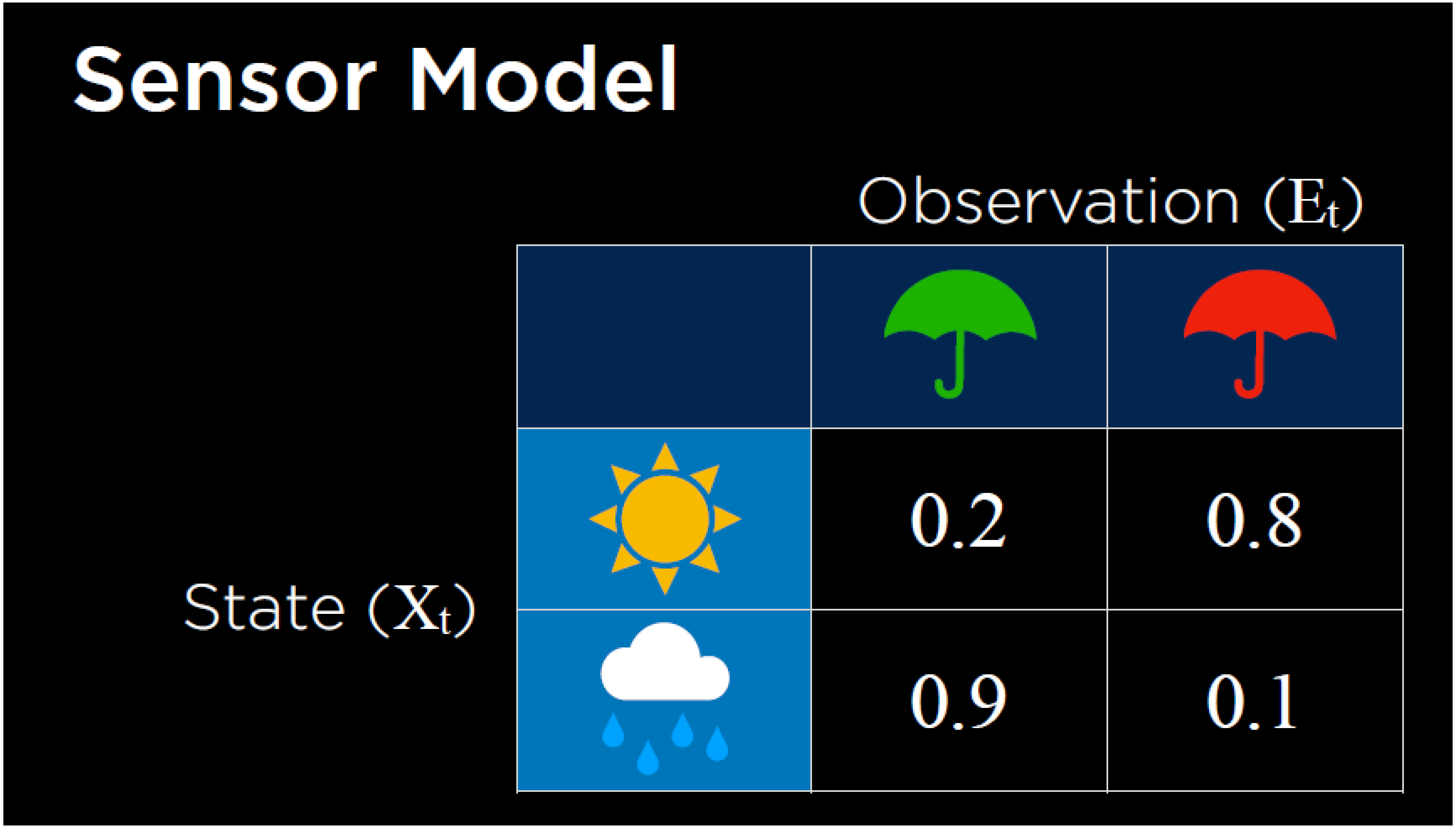


Hidden Markov Model

a Markov model for a system with hidden states that generate some observed event

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

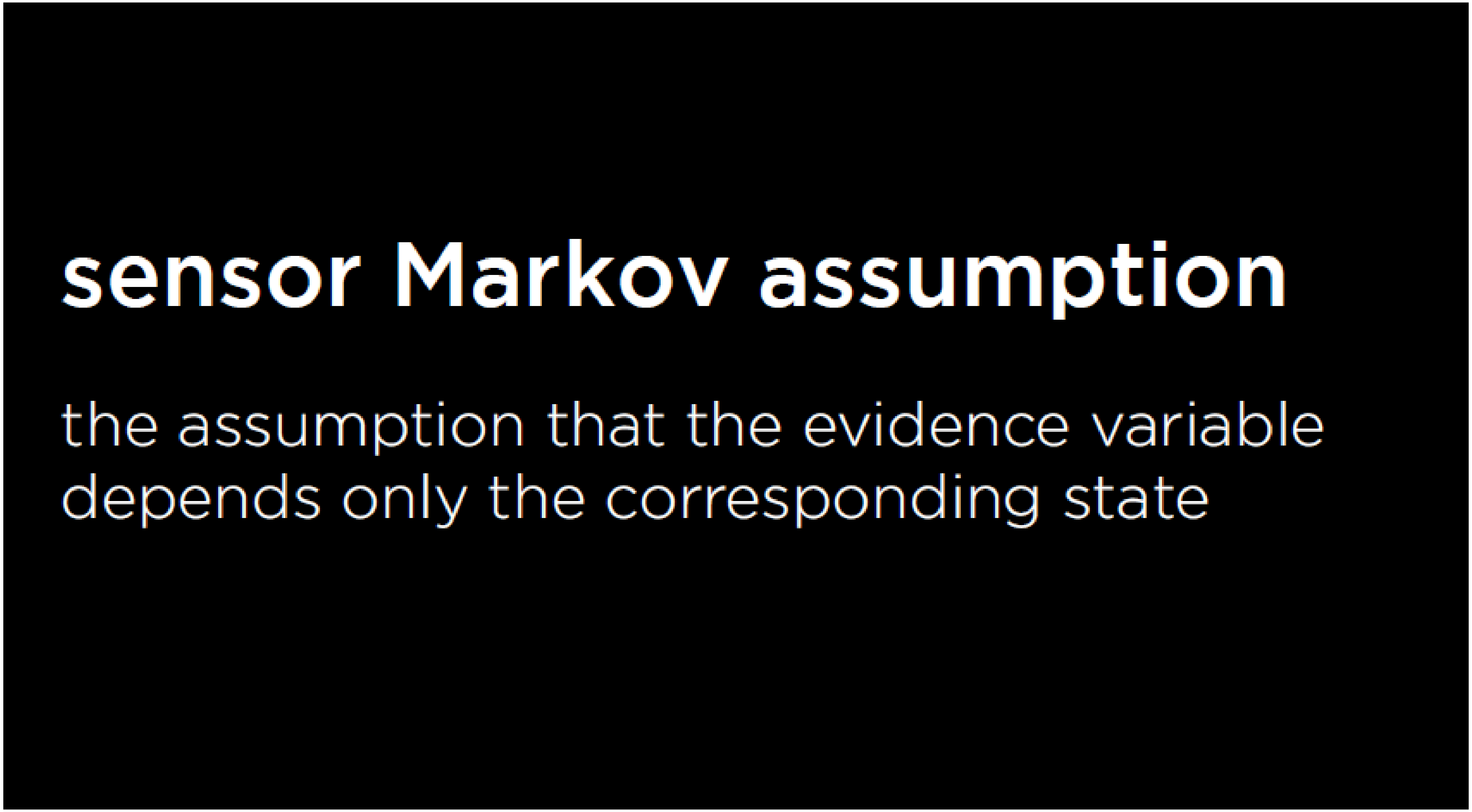


Sensor Model

		Observation (E_t)	
		🟩☂	🟥☂
State (X_t)	☀	0.2	0.8
	🌧	0.9	0.1

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

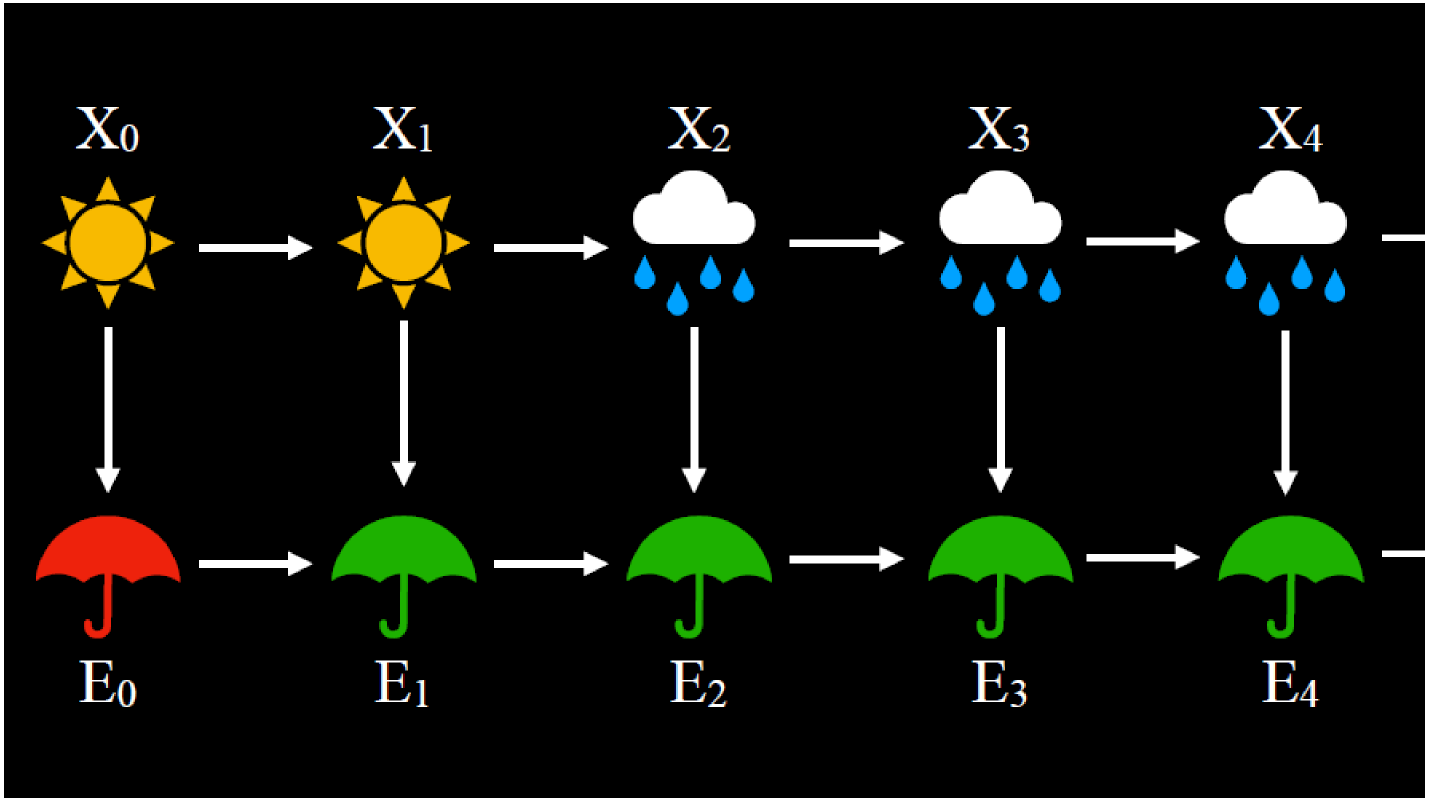


sensor Markov assumption

the assumption that the evidence variable depends only the corresponding state

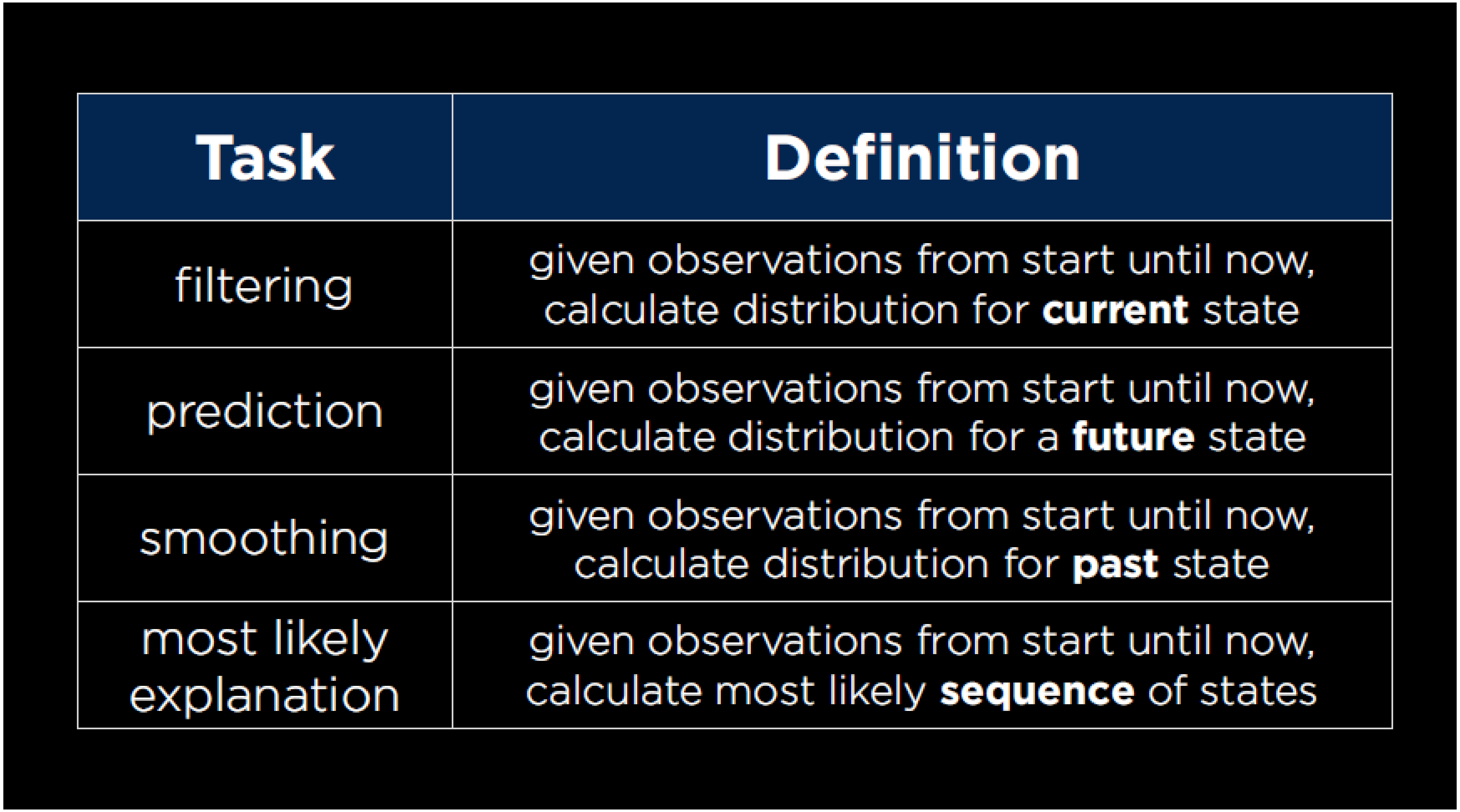
Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time



Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time



Task	Definition
filtering	given observations from start until now, calculate distribution for **current** state
prediction	given observations from start until now, calculate distribution for a **future** state
smoothing	given observations from start until now, calculate distribution for **past** state
most likely explanation	given observations from start until now, calculate most likely **sequence** of states

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python, David J. Malan and Brian Yu

Uncertainty over time

Slide credit : HarvardX CS50AICS50's Introduction to Artificial Intelligence with Python,
David J. Malan and Brian Yu