

# Artificial Intelligence

Algorithms and Applications with Python

Chapter 8



Dr. Dominik Jung

[dominik.jung42@gmail.com](mailto:dominik.jung42@gmail.com)



python

# Outline

8

## Probabilistic Reasoning and Modelling

8.1

From Uncertainty to Probability

8.2

Probabilistic Reasoning

8.3

Probabilistic Reasoning over Time

8.4

Decision Theory and Decision-Making

8.5

Game Theory and Sequential Decision-Making

Lectorial 6: Intelligent Agents in Action

### ► What you will learn:

- Concepts of statistics and probability theory to model agents that can act under uncertainty
- Foundations of Markov theory, Bayes theory and sequence analysis for to reason under uncertainty according to the laws of probability theory
- How to analyze and build optimal agent decision-making that can handle uncertainty

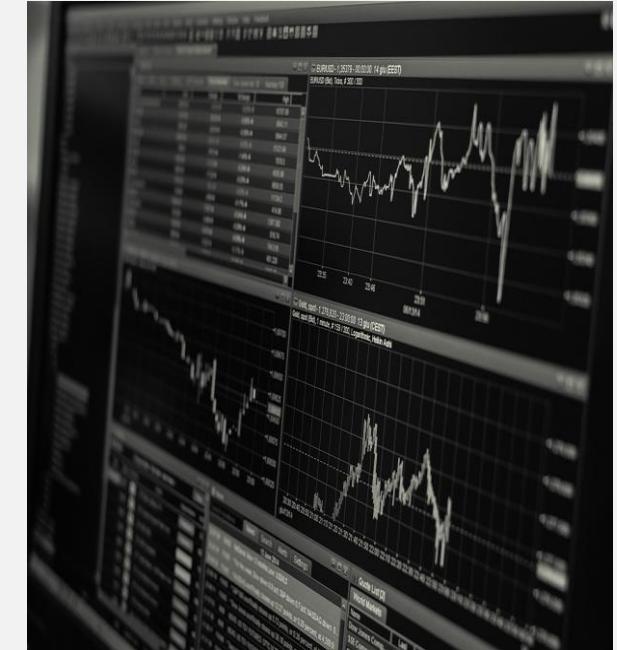


Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

### ► Duration:

- 215 min + 90 min (Lectorial)

### ► Relevant for Exam:

- 8.1 – 8.4

## 8.1 Problems of Previous Agent Implementations

- So far, agents keep a track of a belief state, and generating a plan to act under every possible situation
- Let action  $A_t = \text{time to reach visitor center}$ 
  - Will  $A_t$  get me there on time?
- **Problems of Uncertainty:**
  - Partial observability (road state, other drivers' plans)
  - Noisy sensors (traffic reports)
  - Uncertainty in action outcomes (flat tire, etc.)
  - Complexity of modeling and predicting traffic

→ Russel & Norvig: Laziness and Ignorance dilemma

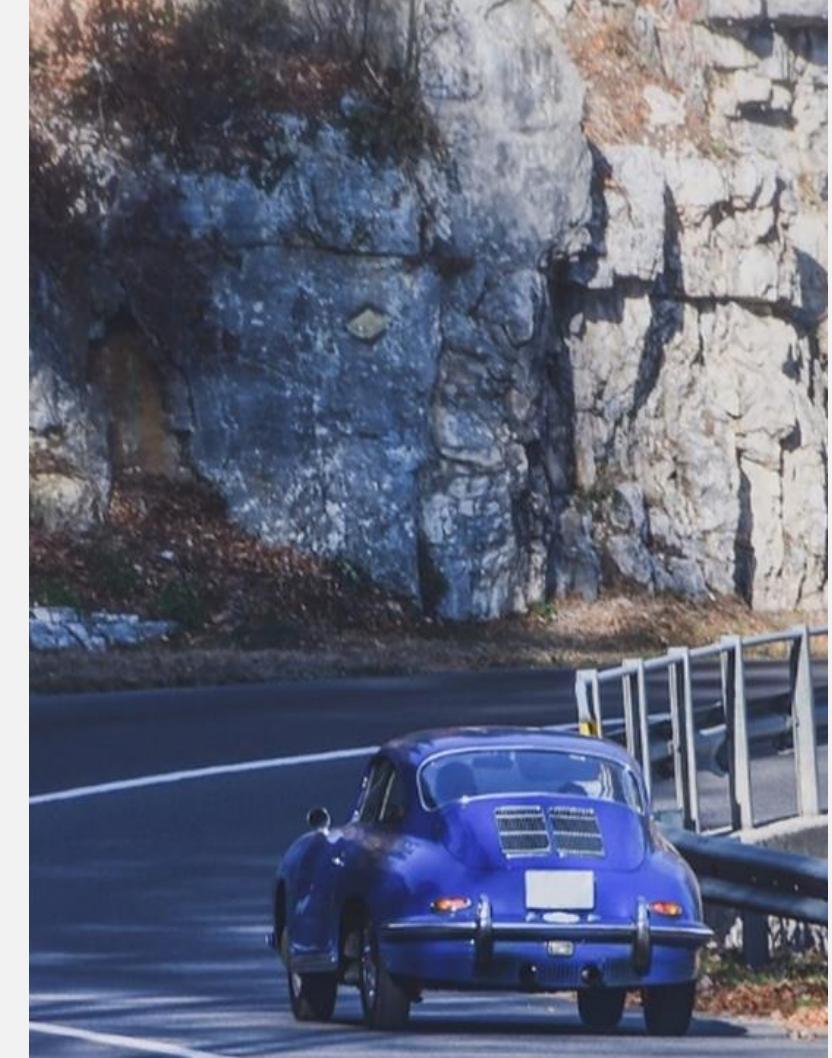


Image source: ↗ Pixabay (2019) / ↗ CCO

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Motivation: Probabilistic Reasoning in Artificial Intelligence

- Hence a purely logical approach either risks falsehood:
  - “ $A_{120}$  will get me there on time”
- Or, leads to conclusions that are too weak for decision making:
  - $A_{120}$  will get us there on time if there's no accident on the way and it doesn't rain and my tires remain intact,...
  - $A_{12000}$  might reasonably be said to get us there on time but I'd have to stay overnight on the road before the next national park
- **Solution:** Agents may need to handle uncertainty, whether due to partial observability, nondeterminism, or a combination of both.

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.1 Uncertainty

- Suppose the agent believes the following:

$P(A_{120} \text{ gets me there on time}) = 5\%$

$P(A_{240} \text{ gets me there on time}) = 60\%$

$P(A_{300} \text{ gets me there on time}) = 95\%$

$P(A_{600} \text{ gets me there on time}) = 99.9\%$

$P(A_{1000} \text{ gets me there on time}) = 99.99\%$



Which action should the agent choose? What is “the best” action if you have to decide for one of these during your national park roadtrip?

## 8.1 Decision-Making under Uncertainty

- Which action should the agent choose?
  - Depends on preferences for missing a national park vs. time spent waiting
  - Encapsulated by a utility function
- **Utility:** Some states may be desirable, others may be undesirable
- Many environments are uncertain in the sense that it is not clear what state an action will lead to
- **Uncertainty:** Some states may be likely, others may be unlikely

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Decision-Making under Uncertainty to Decision Theory

- The agent should choose the action that maximizes the expected utility:

$$P(\text{Success} \mid A_t) \cdot U(\text{Success} \mid A_t) + P(\text{Fail} \mid A_t) \cdot U(\text{Fail} \mid A_t)$$

- Utility theory is used to represent and infer preferences
- Hence,

*Decision theory = probability theory + utility theory*

## 8.1 Decision-Theoretic Agent

- ▶ A decision-theoretic agent that selects rational actions based on decision and utility theory.

- Comparable to previous agents that maintain a belief state reflecting the history of percepts to date
- Difference is that the DT-agent's belief state represents not just the possibilities for world states but also their probabilities
- Given the belief state, the agent can make probabilistic predictions of action outcomes to select the action with highest expected utility.

### Algorithm: DT-Agent

```
function DT-AGENT(percept) returns an action  
persistent  
    belief_state, probabilistic beliefs about the current  
    state of the world  
    action, the agent's action  
  
    update belief_state based on action and percept  
    calculate outcome probabilities for actions,  
        given action descriptions and current belief_state  
    select action with highest expected utility  
        given probabilities of outcomes and utility  
        information  
  
return action
```

Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 What is Probability?

### Frequentism/Objectivism

- Probabilities are relative frequencies, that come from repeated experiments
- For example, if we flip a coin many times, we see experimentally that it comes out heads  $\frac{1}{2}$  the time
- For most events in the world, there is no history of exactly that event happening
- “Reference class” problem



### Subjectivism

- Probabilities merely reflect agents' beliefs
- But then, how do we assign belief values to statements? And what would constrain agents to hold consistent beliefs
- “Assignment” problem



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020) | Image source: ↗ Pixabay (2019) / ↗ CCO

## 8.1 Random Variables

- We describe the (uncertain) state of the world using random variables

*S: What is the speed of my porsche?\**

- Just like variables in constrain-satisfaction problems (CSP), random variables take on values in a domain (domain values must be mutually exclusive and exhaustive)

$$S \text{ in } [0, 350] \frac{\text{km}}{\text{h}}$$

## 8.1 Events and Propositions

- Probabilistic statements are defined over events, or sets of world states

*"On the A8 I can drive between 250 and 350 km/h"*

- Events are described using propositions

$$250 \leq S < 350$$

## 8.1 Notion of Probability (P-Notation)

- Hence:

$$P(\text{proposition}) = x$$

- $P(A)$  is the probability of the set of world states in which proposition  $A$  holds
- $P(X = x)$ , or  $P(x)$  for short, is the probability that random variable  $X$  has taken on the value  $x$

## 8.1 Kolmogorov's Axioms of Probability

D

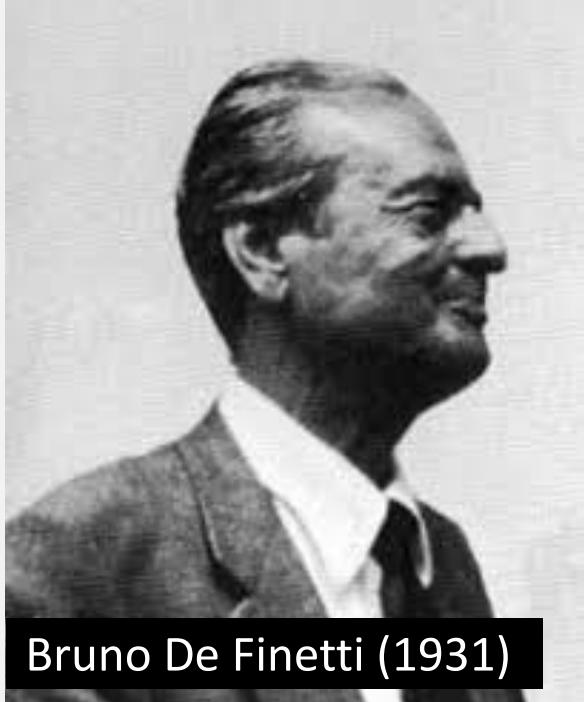
For any propositions (events) A, B

- $0 \leq P(A) \leq 1$
- $P(\text{True}) = 1$  and  $P(\text{False}) = 0$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- These axioms are sufficient to completely specify probability theory for discrete random variables (for continuous variables, we need density functions)



Based on these axioms,  
can you compute  $P(\neg A)$ ?

## 8.1 Probabilities and Rationality



Bruno De Finetti (1931)

Image source: ↗ [Bruno de Finetti](#) (2019) by Ffaffff from ↗ [Wikimedia](#)

- Why should a rational agent hold beliefs that are consistent with axioms of probability?
- De Finetti postulates, if an agent has some degree of belief in proposition  $A$ , it should be able to decide whether or not to accept a bet for/against  $A$
- If the agent believes that  $P(A) = 0.4$ , should it agree to bet 60€ that  $A$  will occur against 40€ that  $A$  will not occur?



**D** Finetti Theorem: An agent who holds beliefs inconsistent with axioms of probability can be tricked into accepting a combination of bets that are guaranteed to lose them money

Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 Why we need Kolmogorov's Axioms of Probability in AI



- Given an agent 1 with the following beliefs

$$P(A) = 0.3, \quad P(B) = 0.2, \quad P(A \vee B) = 0.9$$

- E.g. smart agent 2 bets the opposite of agent 1 in each game and bets 7MU on  $A$ , 8MU on  $B$ , and 9MU on  $\neg(A \vee B)$

→ Agent 1 loses money

## 8.1 Variables and Atomic Events

- If the world would consist of only of the following two Boolean variables then there are four related distinct atomic events:

*Warning signal  
Engine broken*

$\text{Broken} = \text{false} \wedge \text{Warning} = \text{false}$   
 $\text{Broken} = \text{false} \wedge \text{Warning} = \text{true}$   
 $\text{Broken} = \text{true} \wedge \text{Warning} = \text{false}$   
 $\text{Broken} = \text{true} \wedge \text{Warning} = \text{true}$

- **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



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

## 8.1 Joint Probability Distributions

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

Atomic Event	Probability
$\text{Broken} = \text{false} \wedge \text{Warning} = \text{false}$	0.8
$\text{Broken} = \text{false} \wedge \text{Warning} = \text{true}$	0.1
$\text{Broken} = \text{true} \wedge \text{Warning} = \text{false}$	0.05
$\text{Broken} = \text{true} \wedge \text{Warning} = \text{true}$	0.05



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

## 8.1 Marginal Probability Distributions

- Suppose we have the joint distribution  $P(X, Y)$  and we want to find the marginal distribution  $P(Y)$

P(Broken, Warning)	
$\text{Broken} = \text{false} \wedge \text{Warning} = \text{false}$	0.8
$\text{Broken} = \text{false} \wedge \text{Warning} = \text{true}$	0.1
$\text{Broken} = \text{true} \wedge \text{Warning} = \text{false}$	0.05
$\text{Broken} = \text{true} \wedge \text{Warning} = \text{true}$	0.05

P(Warning)	
$\text{Warning} = \text{false}$	?
$\text{Warning} = \text{true}$	?

P(Broken)	
$\text{Broken} = \text{false}$	?
$\text{Broken} = \text{true}$	?

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Marginal Probability Distributions

- Suppose we have the joint distribution  $P(X, Y)$  and we want to find the marginal distribution  $P(X)$

$$\begin{aligned} P(X = x) &= P((X = x \wedge Y = y_1) \vee \dots \vee (X = x \wedge Y = y_n)) \\ &= P((x, y_1) \vee \dots \vee (x, y_n)) = \sum_{i=1}^n P(x, y_i) \end{aligned}$$

- General rule:** to find  $P(X = x)$ , sum the probabilities of all atomic events where  $X = x$ .

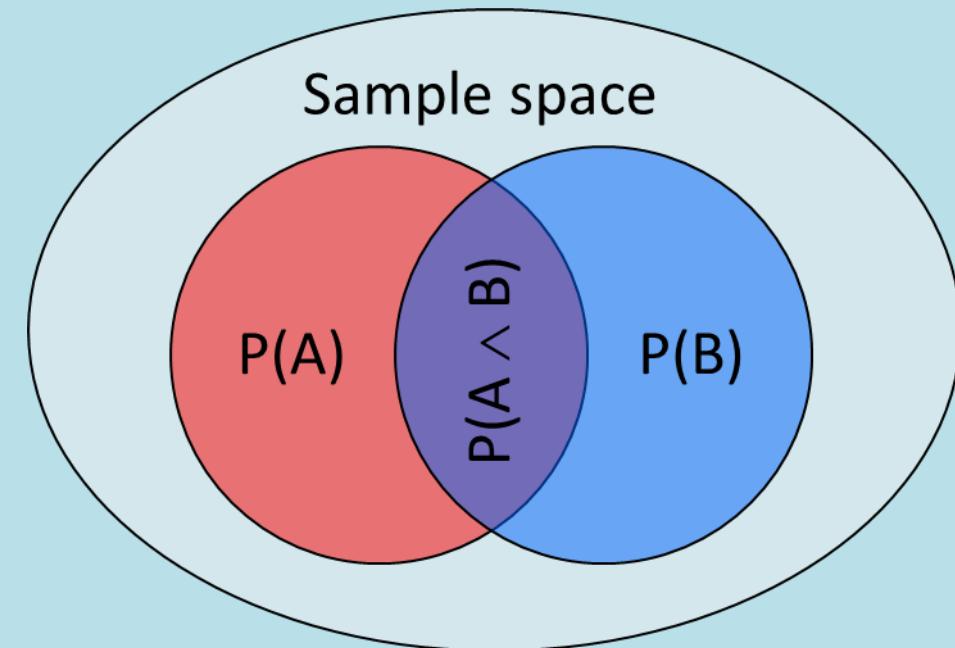
## 8.1 Conditional Probability Visualization

- Probability of warning given the engine is really broken

$$P(\text{Broken} = \text{true} \mid \text{Warning} = \text{true})$$

- For any two events A and B,

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



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Conditional Probability

P (Broken, Warning)	
<i>Broken = false</i> $\wedge$ <i>Warning = false</i>	0.8
<i>Broken = false</i> $\wedge$ <i>Warning = true</i>	0.1
<i>Broken = true</i> $\wedge$ <i>Warning = false</i>	0.05
<i>Broken = true</i> $\wedge$ <i>Warning = true</i>	0.05

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

P (Warning)	
<i>Warning = false</i>	0.85
<i>Warning = true</i>	0.15

P (Broken)	
<i>Broken = false</i>	0.9
<i>Broken = true</i>	0.1

- $P(\text{Broken} = \text{true} | \text{Warning} = \text{false}) = \frac{0.05}{0.85} = 0.059$
- $P(\text{Broken} = \text{false} | \text{Warning} = \text{true}) = \frac{0.1}{0.15} = 0.667$

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Conditional Distributions

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

$P(Broken, Warning)$	
$Broken = false \wedge Warning = false$	0.8
$Broken = false \wedge Warning = true$	0.1
$Broken = true \wedge Warning = false$	0.05
$Broken = true \wedge Warning = true$	0.05

$P(Warning | Broken = true)$

$Warning = false$	0.5
$Warning = true$	0.5

$P(Warning | Broken = false)$

$Broken = false$	0.889
$Broken = true$	0.111

$P(Broken | Warning = true)$

$Broken = false$	0.667
$Broken = true$	0.333

$P(Broken | Warning = false)$

$Broken = false$	0.941
$Broken = true$	0.059



Do you have any idea how to calculate these conditional distributions at once?

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Normalization Trick

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

$P(Broken, Warning)$	
$Broken = \text{false} \wedge \text{Warning} = \text{false}$	0.8
$Broken = \text{false} \wedge \text{Warning} = \text{true}$	0.1
$Broken = \text{true} \wedge \text{Warning} = \text{false}$	0.05
$Broken = \text{true} \wedge \text{Warning} = \text{true}$	0.05

**D Renormalization**

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

↓ Select

$Warning, Broken = \text{false}$	
$Warning = \text{false}$	0.8
$Warning = \text{true}$	0.1

Renormalize →

$P(Warning   Broken = \text{false})$	
$Broken = \text{false}$	0.889
$Broken = \text{true}$	0.111

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Bayes Rule

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

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

- Therefore



**Bayes Rule**

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

- Agent can get diagnostic probability  $P(\text{Broken} | \text{Warning})$  from causal probability  $P(\text{Warning} | \text{Broken})$

- Agent can now update its beliefs based on evidence

- In general, important tool for probabilistic inference

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020)

## 8.1 Statistical Independence

- Two events are independent, statistically independent if the occurrence of one does not affect the probability of occurrence of the other

**D**

Two events  $A$  and  $B$  are independent if and only if:

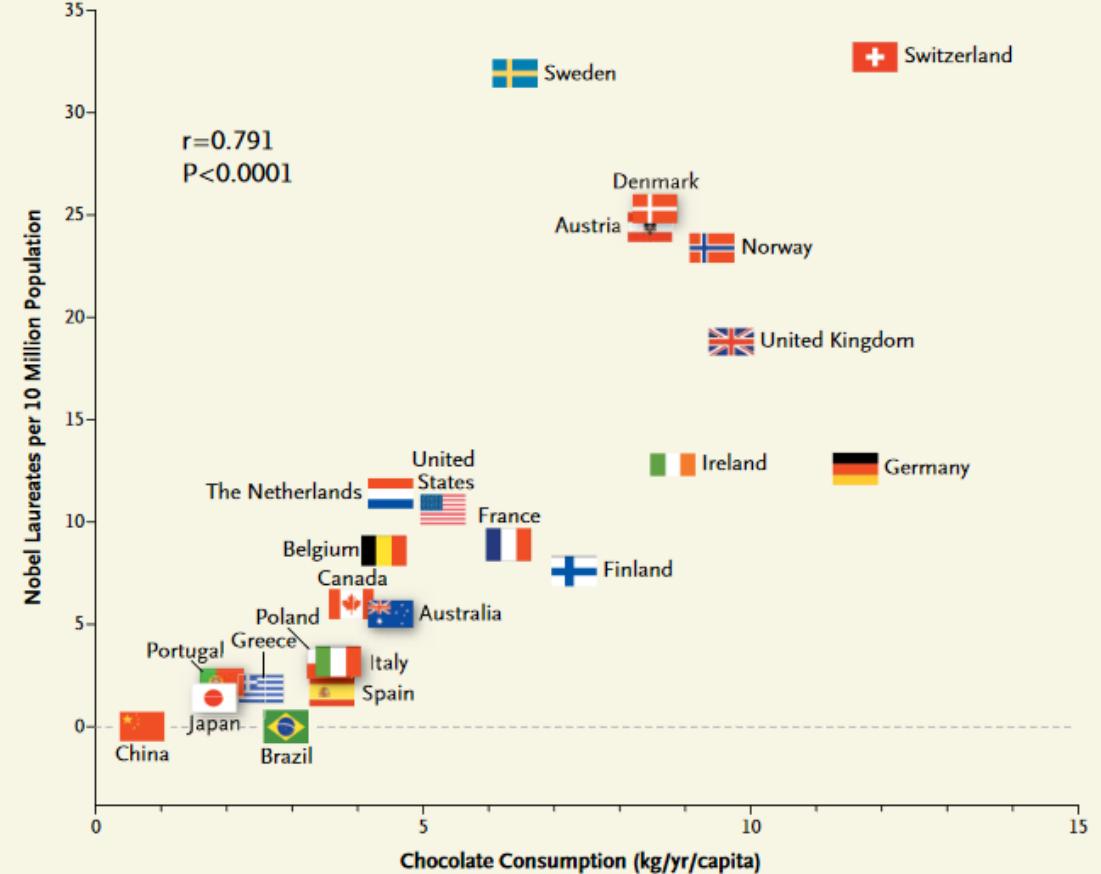
$$P(A \wedge B) = P(A)P(B)$$

- In other words

$$P(A | B) = \\ P(A) \text{ and } P(B | A) = P(B)$$

- Causality vs. Correlation

Correlation between chocolate consumption and Nobel prizes



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020); Image Source: Messerli, F. H. et al. (2012)

## 8.1 Conditional Independence: Example I

- If the car's engine is broken, the probability that the car stops doesn't depend on whether there is a warning message or not

$$P(\text{Stop} \mid \text{Warning}, \text{Broken}) = P(\text{Stop} \mid \text{Broken})$$

- Therefore, that you have to interrupt your road trip and go to a mechanic is conditionally independent of a warning message given a broken engine

## 8.1 Conditional Independence: Example II

- Likewise, a warning message is conditionally independent of a car stop given a broken engine

$$P(\text{Warning} \mid \text{Broken}, \text{Stop}) = P(\text{Warning} \mid \text{Broken})$$

- Equivalent statement:

$$\begin{aligned} & P(\text{Warning}, \text{Stop} \mid \text{Broken}) \\ &= P(\text{Warning} \mid \text{Broken}) P(\text{Stop} \mid \text{Broken}) \end{aligned}$$



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  $n$

## 8.1 Naïve Bayes Model

- Suppose we have many different types of observations (symptoms, features) that we want to use to diagnose the underlying cause
- It is usually impractical to directly estimate or store the joint distribution

$$P(\text{Cause}, \text{Effect}_1, \dots, \text{Effect}_n).$$

- To simplify things, we can assume that the different effects are conditionally independent given the underlying cause

## 8.1 Naïve Bayes Model

- To simplify things, we can assume that the different effects are conditionally independent given the underlying cause
- Then we can estimate the joint distribution as

$$P(\text{Cause}, \text{Effect}_1, \dots, \text{Effect}_n).$$

$$P(\text{Cause}, \text{Effect}_1, \dots, \text{Effect}_n) = P(\text{Cause}) \prod_i P(\text{Effect}_i | \text{Cause})$$

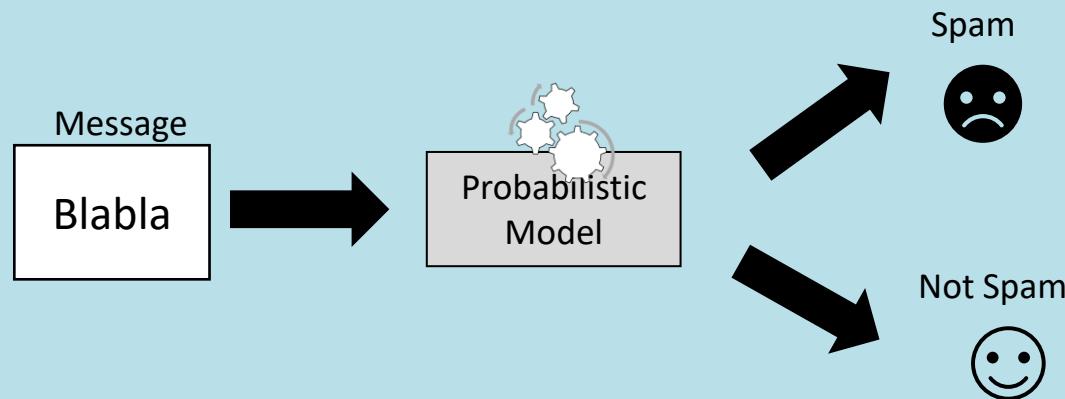
- This is usually not accurate, but very useful in practice ☺

## 8.1 Example: Naïve Bayes Spam Filter

- **Bayesian decision theory:** to minimize the probability of error, we should classify a message as spam if

$$P(\text{spam} \mid \text{message}) > P(\neg \text{spam} \mid \text{message})$$

- Maximum a posteriori (MAP) decision



Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 Example: Naïve Bayes Spam Filter

### ■ Spam Classification

$$P(\text{spam} \mid w_1, \dots, w_n) = P(\text{spam}) \prod_{i=1}^n P(w_i \mid \text{spam})$$

posterior              prior              likelihood

spam: 0.33 ¬spam: 0.67	word1: 0.002 word2: 0.06 ...  word1: 0.08 word2: 0.03 ...	$P(\text{word} \mid \text{spam})$ $P(\text{word} \mid \neg\text{spam})$
---------------------------	---	--

Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 Example: Naïve Bayes Spam Filter

- We have

$$P(\text{spam} | \text{message}) = \frac{P(\text{message} | \text{spam})P(\text{spam})}{P(\text{message})} \quad \text{and}$$

$$P(\neg \text{spam} | \text{message}) = \frac{P(\text{message} | \neg \text{spam})P(\neg \text{spam})}{P(\text{message})}$$

- Notice that  $P(\text{message})$  is just a constant normalizing factor and doesn't affect the decision
- Therefore, all we need is to find  $P(\text{message} | \text{spam}) P(\text{spam})$  and  $P(\text{message} | \neg \text{spam}) P(\neg \text{spam})$

## 8.1 Example: Naïve Bayes Spam Filter

- We need to find  $P(\text{message} \mid \text{spam}) P(\text{spam})$  and  $P(\text{message} \mid \neg \text{spam}) P(\neg \text{spam})$
- The message is a sequence of words  $(w_1, \dots, w_n)$
- **Bag of words** representation
  - The order of the words in the message is not important
  - Each word is conditionally independent of the others given message class (spam or not spam)

$$P(\text{message} \mid \text{spam}) = P(w_1, \dots, w_n \mid \text{spam}) = \prod_{i=1}^n P(w_i \mid \text{spam})$$

- Our classification model will classify the message as spam if

$$P(\text{spam}) \prod_{i=1}^n P(w_i \mid \text{spam}) > P(\neg \text{spam}) \prod_{i=1}^n P(w_i \mid \neg \text{spam})$$

Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 Example: Simple Perceptron in Python

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter naive bayes spam classification (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Status bar: Kernel starting, please wait..., Saving every 120 sec., Not Trusted, Python 3.
- Code Cell:** In [ ]: 

```
import pandas as pd
```
- Text Cell:** Let us start to take a look at the spam dataset
- File Icon:** A circular icon with a white document symbol.
- File Path:** code/lecture 8- naive bayes spam classification.ipynb

# Coding Session



- **Application for us:** In general, the agent observes the values of some random variables  $X_1, X_2, \dots, X_n$  and needs to reason about the values of some other unobserved random variables  $Y_1, Y_2, \dots, Y_m$
- **General applications:**
  - Figuring out a diagnosis based on symptoms and test results
  - Classifying the content type of an image or a document based on some features
  - Etc.

## 8.1 Sources of Probabilities

- Statistical Experiments
- Knowledge or Experience
- Estimating
- What is the probability that the sun will rise tomorrow?

Adapted from Russell, S., & Norvig, P. (2016)

## 8.1 Classroom task



Previous  
Exam Task!

# Your turn!

### Task

Please answer the following questions:

- How many entries do we need to represent the joint probability table of  $P(\text{Warning}, \text{Broken}, \text{Stop})$ ?
- Write out the joint distribution using chain rule:  $P(\text{Warning}, \text{Broken}, \text{Stop})$ . How many entries do we need to represent these distributions?
- Where is the benefit of the usage of conditional independence in this example? Why do AI scientists do that?

# Outline

8

## Probabilistic Reasoning and Modelling

8.1

From Uncertainty to Probability

8.2

Probabilistic Reasoning

8.3

Probabilistic Reasoning over Time

8.4

Decision Theory and Decision-Making

8.5

Game Theory and Sequential Decision-Making

Lectorial 6: Intelligent Agents in Action

► **What you will learn:**

- Concepts of statistics and probability theory to model agents that can act under uncertainty
- Foundations of Markov theory, Bayes theory and sequence analysis for to reason under uncertainty according to the laws of probability theory
- How to analyze and build optimal agent decision-making that can handle uncertainty

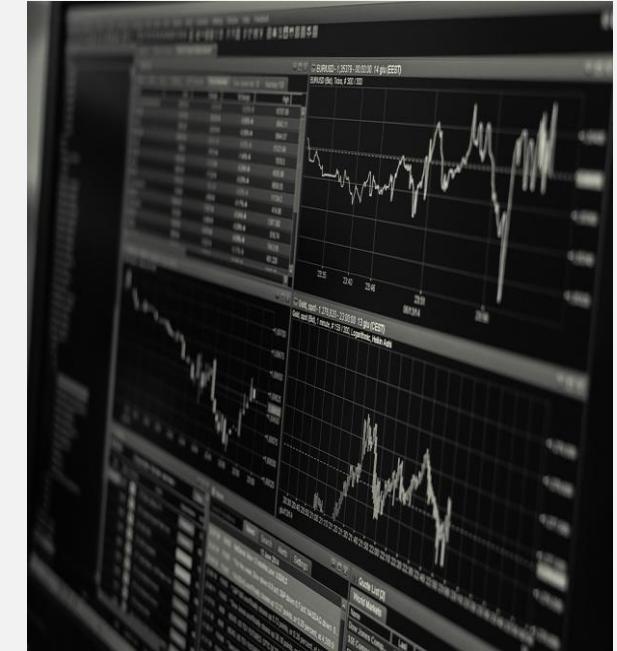


Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

► **Duration:**

- 215 min + 90 min (Lectorial)

► **Relevant for Exam:**

- 8.1 – 8.4

## 8.2 Motivation Base Rates and Probabilistic Inference

*Linda is thirty-one years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations...*

Please rank in order of likelihood the following scenarios: Linda is

- (1) an elementary school teacher,
- (2) active in the feminist movement,
- (3) a bank teller,
- (4) an insurance salesperson, or
- (5) a bank teller also active in the feminist movement.



Amos Tversky,  
Daniel Kahneman,

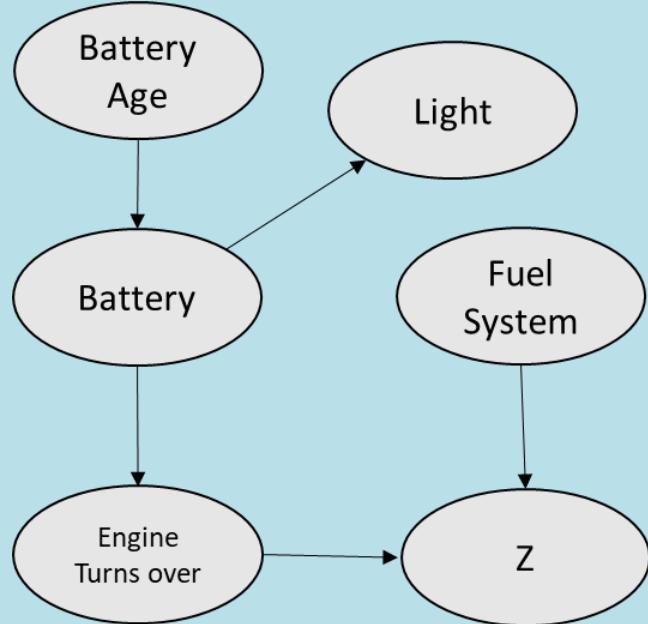
- Given:
  - Query variables:  $X$
  - Evidence (observed) variables:  $E = e$
  - Unobserved variables:  $Y$
- If we know the full joint distribution  $P(X, E, Y)$ , how can we perform inference about  $X$ ?

$$P(X | E = e) = \frac{P(X, e)}{P(e)} \propto \sum_y P(X, e, y)$$

- Problems
  - Full joint distributions are too large
  - Marginalizing out  $Y$  may involve too many summation terms

## 8.2 Bayesian Networks

- Represents a set of variables and their conditional dependencies via a directed acyclic graph



### Example

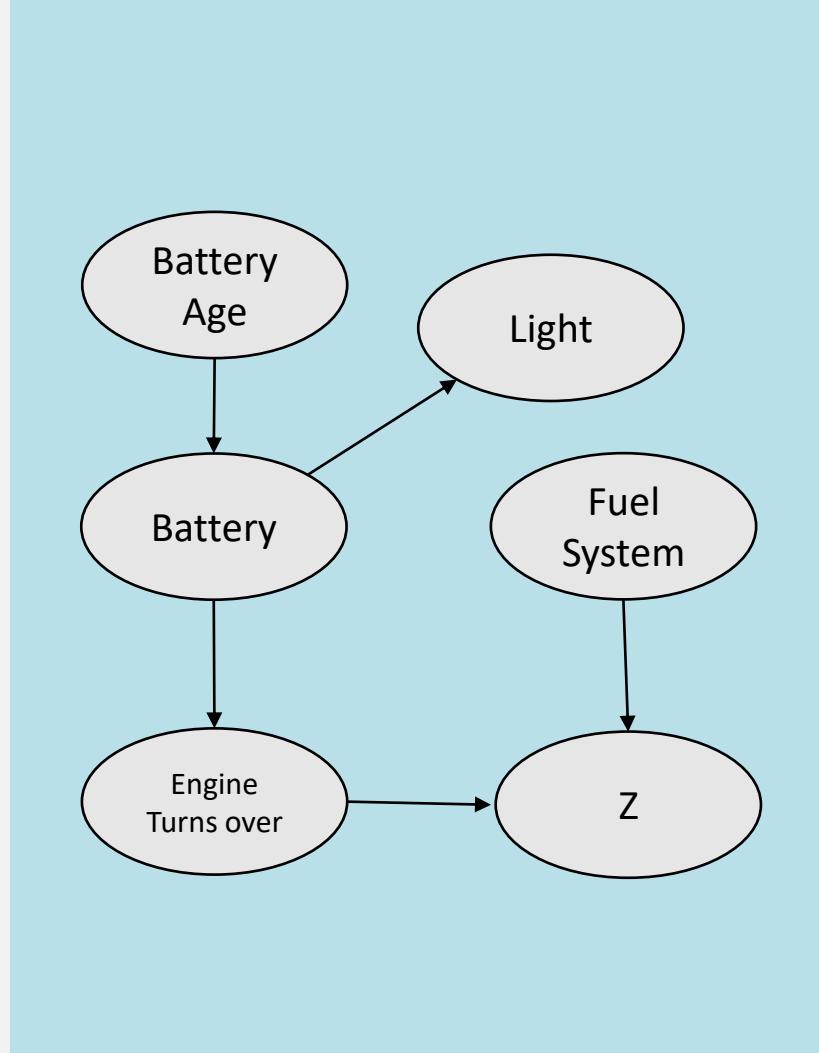
- Root Cause Analytics (RCA) in Automotive Industry

- More commonly called belief or probabilistic networks
- A way to depict conditional independence relationships between random variables
- A compact specification of full joint distributions

- + Natural representation for (causally induced) conditional independence, topology and conditional probability tables, generally easy for domain experts to construct
- Requires greater statistical expertise than some other methods, only as useful as this prior knowledge

## 8.2 Structure of Bayesian Networks

- A Bayesian network is made of nodes and arcs
- Each node represents one random variable and can be assigned (observed) or unassigned (unobserved)
- Arcs represent interactions. An arrow from one variable to another indicates direct influence from that variable upon the other variable
- You can encode conditional independence
- Must form a directed, acyclic graph

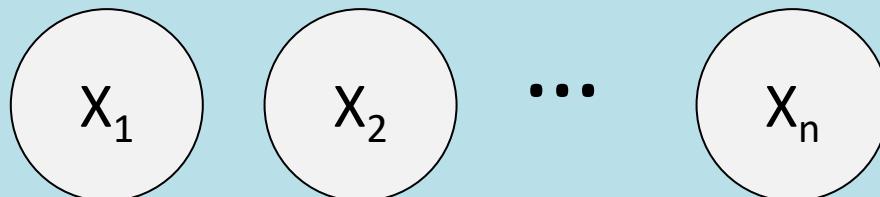


Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Modelling with Bayesian Networks

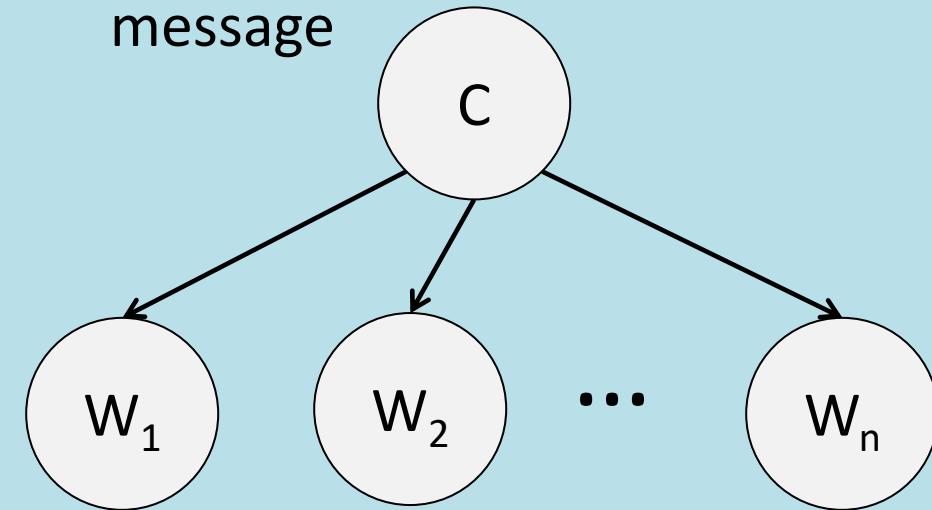
$N$  independent Coin Flips

- Complete independence: no interactions



Naïve Bayes Spam Filter

- Random variables:
  - $C$ : message class (spam or not spam)
  - $W_1, \dots, W_n$ : words comprising the message



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Example: Porsche Theft Protection

- I bought a new Porsche (718) and set up a theft protection alarm that is sometimes set off by animals on the parking space. My two colleagues John and Mary promised to call me if they hear the alarm
  - Example inference task: suppose Mary calls and John doesn't call. Is my Porsche stolen?
- What are the random variables?
  - *Theft, Bird attack on my car, Alarm, John calls, Mary calls*
- Network topology reflects causal knowledge:
  - A thief can set the alarm off
  - A bird shitting on my car's roof can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

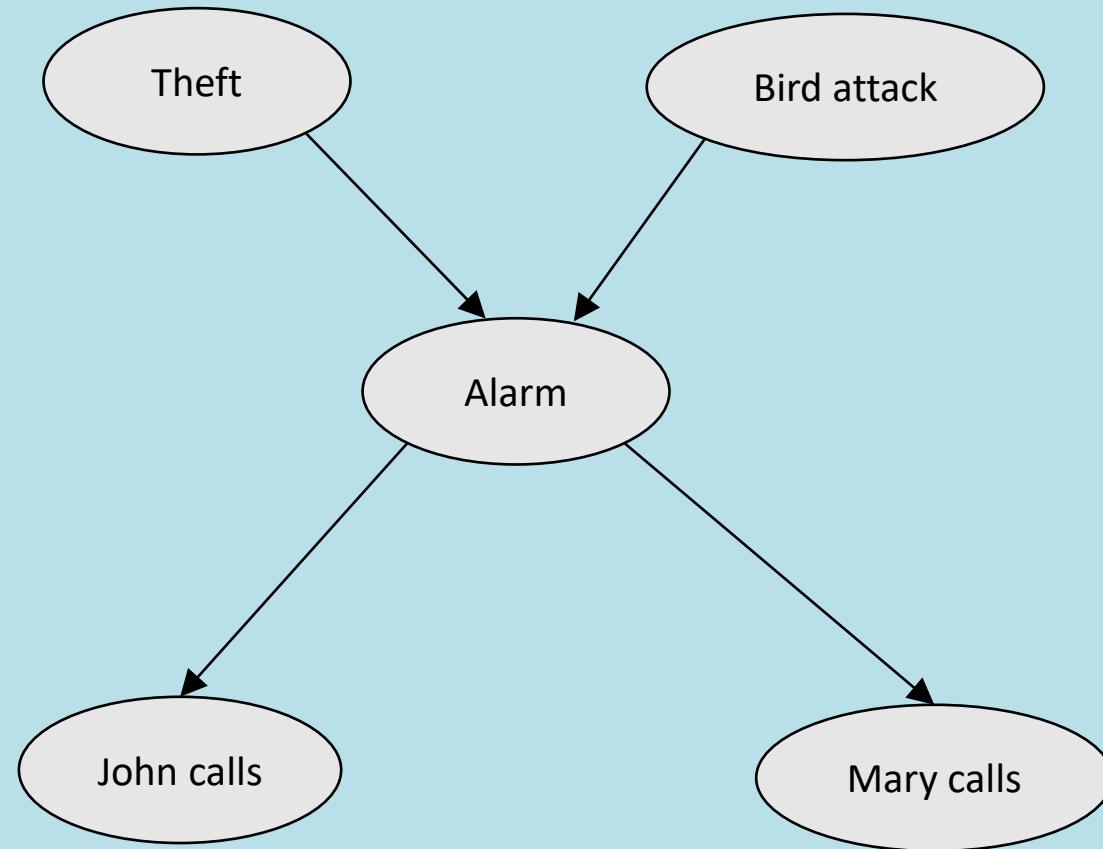


How would the model look like? Why?



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020) | Image source: ↗ Pixabay (2019) / ↗ CCO | The following example is adapted from Judea Pearl who made a similar example with earthquakes and burglaries.

## 8.2 Example: Porsche Theft Protection

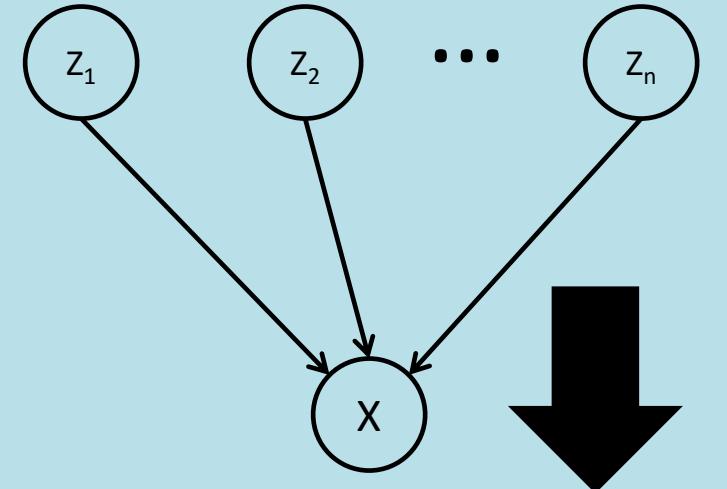


Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Conditional Probability Distributions in Bayesian Networks

- In addition to the structure, we need a conditional probability distribution for each node given the random variables of its parents
- Specify a conditional distribution for each node given its parents:

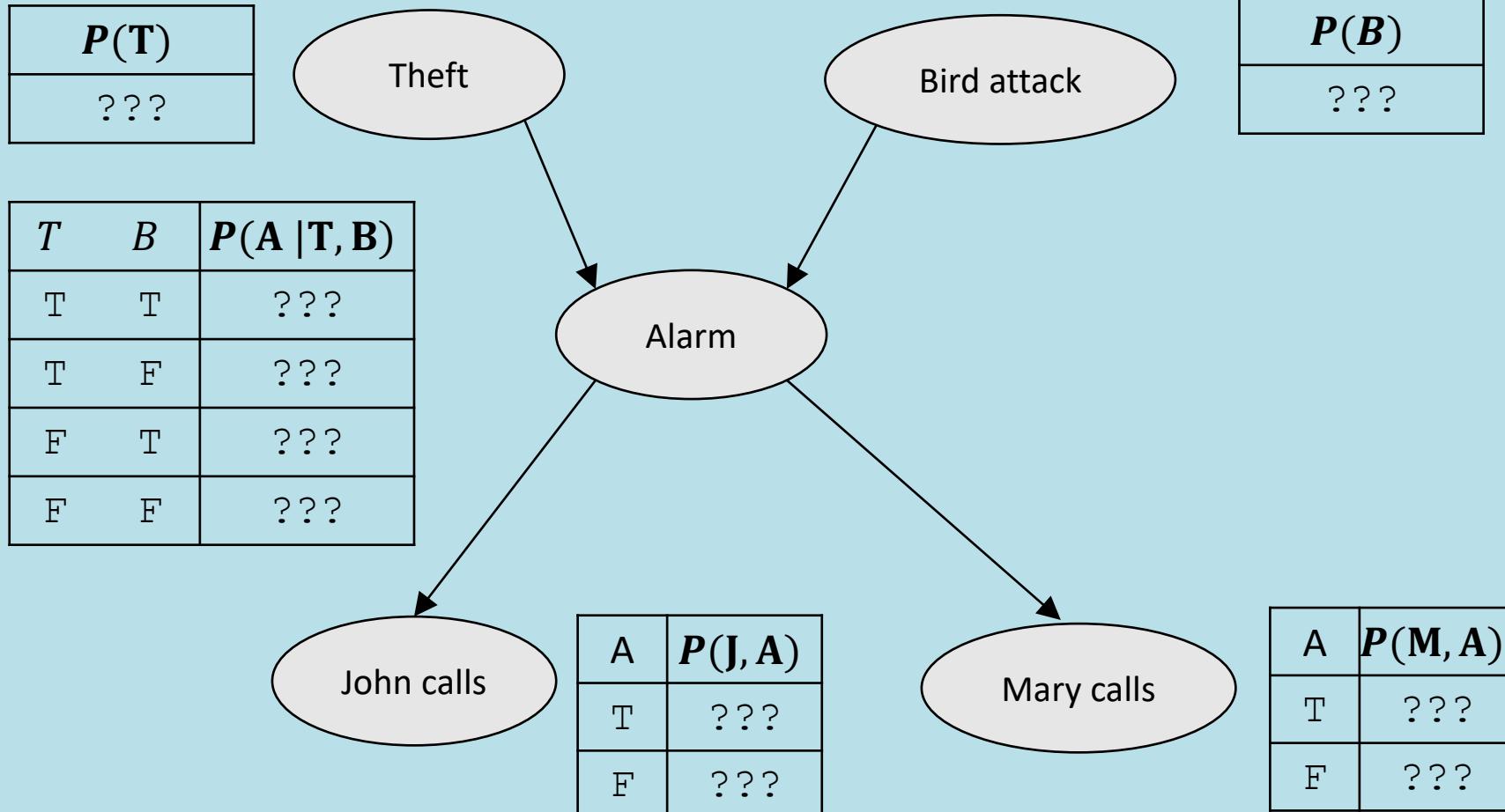
$$P(X | \text{Parents}(X))$$



$$P(X | z_1, \dots, z_n)$$

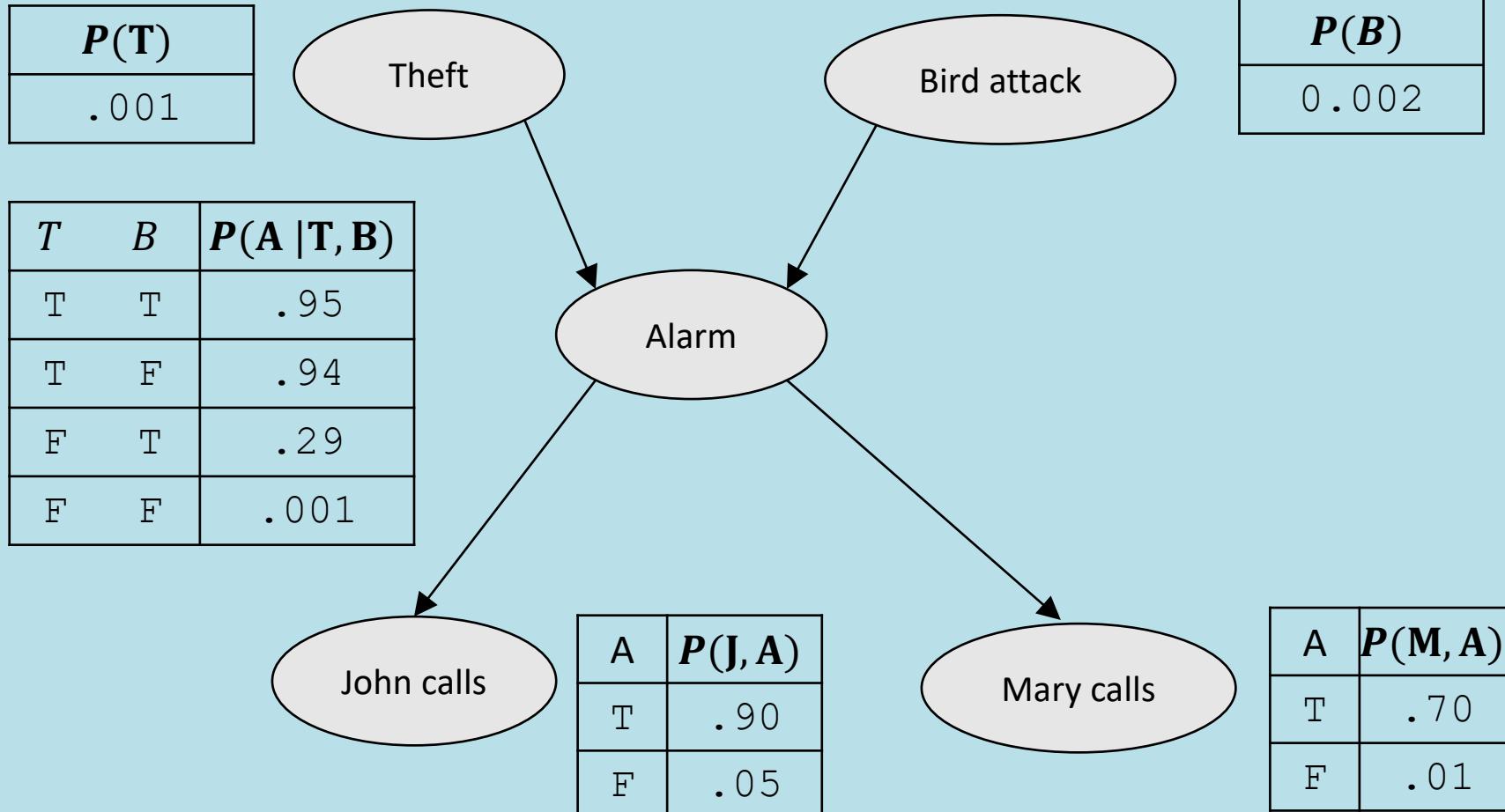
Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Example 1: Porsche Theft Protection I



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Example 1: Porsche Theft Protection II

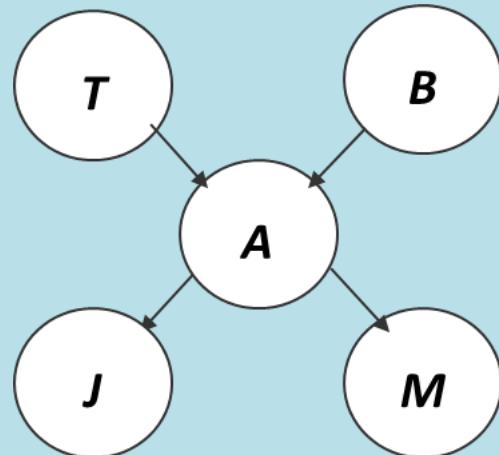


Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 The Joint Probability Distribution

- For each node  $X_i$ , we know  $P(X_i | Parents(X_i))$
- How do we get the full joint distribution  $P(X_1, \dots, X_n)$ ?
- Using chain rule:

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



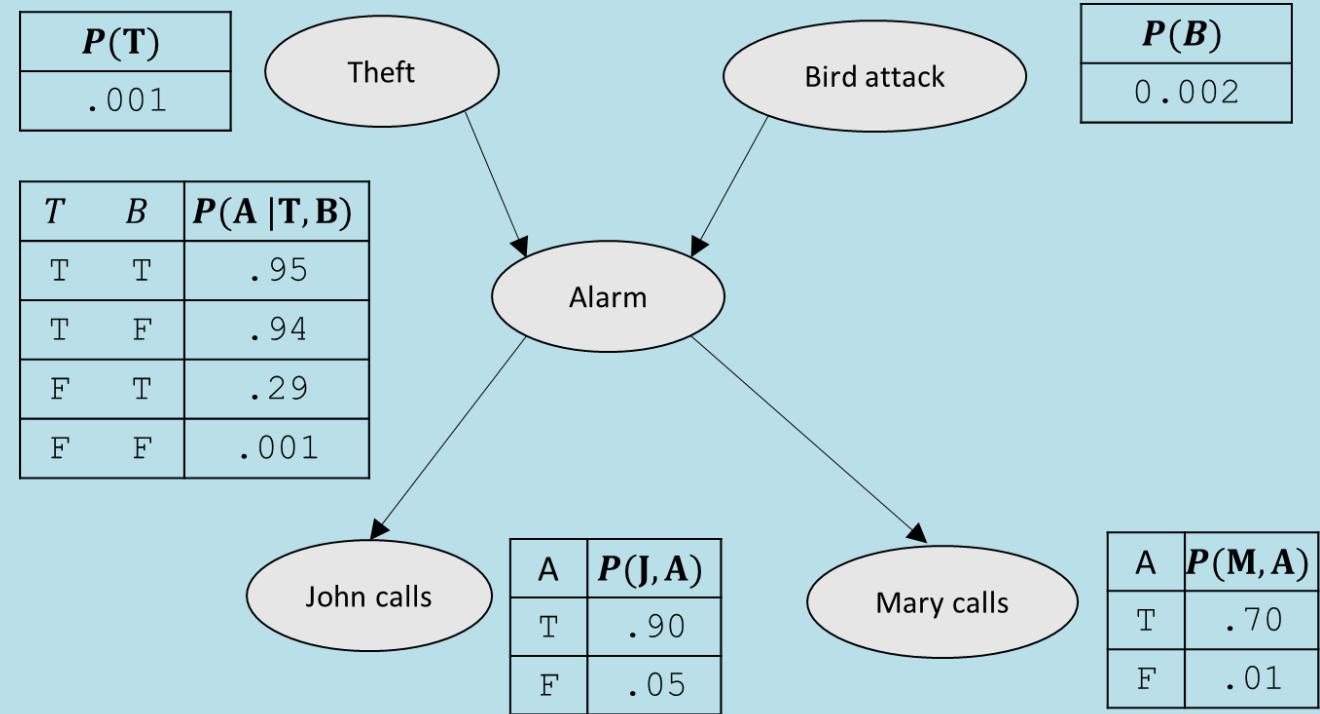
What is the probability that the alarm starts and both John and Mary call, but there is neither a bird attack nor a thief stealing the car?

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Calculation of the Joint Probability Distribution



What is the probability that the alarm starts and both John and Mary call, but there is neither a bird attack nor a thief stealing the car?

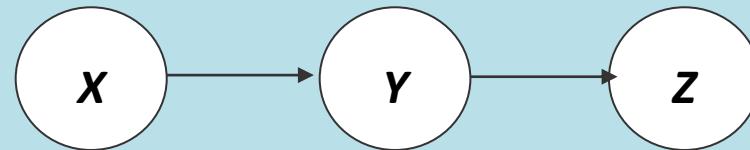


$$\begin{aligned} & \blacksquare P(J, M, A, \neg T, \neg B) \\ &= P(\neg T) P(\neg B) P(A | \neg T, \neg B) P(J | A) P(M | A) \\ &= 0.999 \cdot 0.998 \cdot 0.001 \cdot 0.9 \cdot 0.7 \approx 0.00063 \end{aligned}$$

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Conditional Independence

- Key assumption:  $X$  is conditionally independent of every non-descendant node given its parents
- Example: causal chain



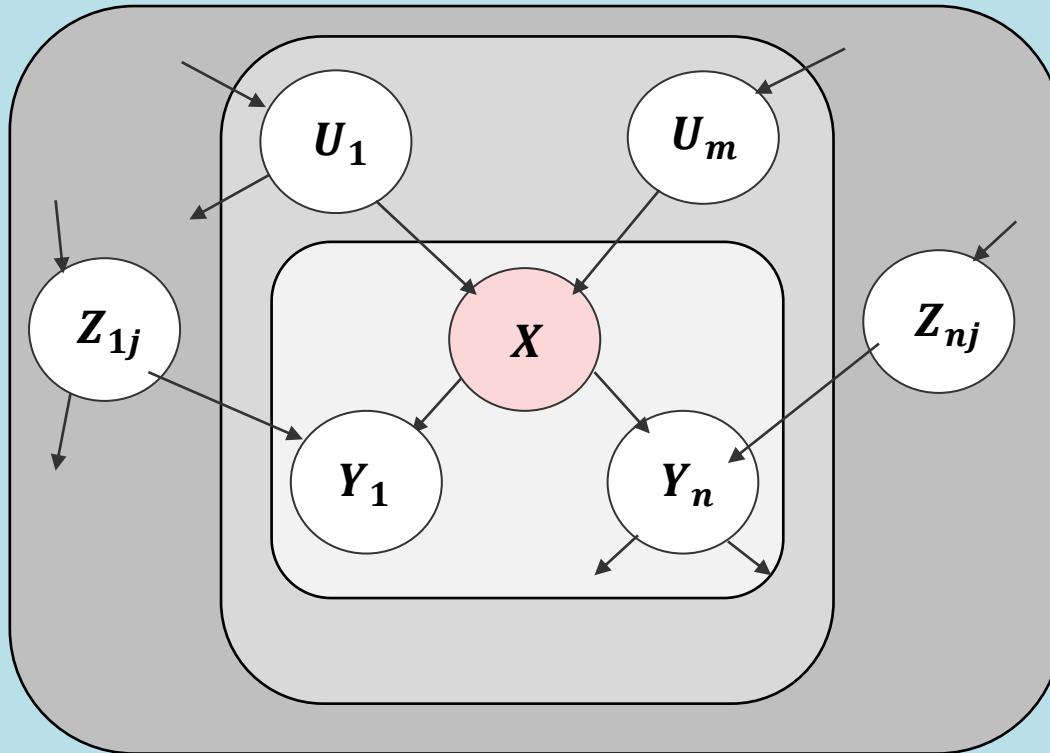
- Are  $X$  and  $Z$  independent?
- Is  $Z$  independent of  $X$  given  $Y$ ?

$$P(Z | X, Y) = \frac{P(X, Y, Z)}{P(X, Y)} = \frac{P(X)P(Y | X)P(Z | Y)}{P(X)P(Y | X)} = P(Z | Y)$$

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Local Semantics of a Bayesian Network

- Each node is conditionally independent of its nondescendants given its parents



$$P(X | U_1, \dots, U_m, Z_{1j}, \dots, Z_{nj}) \\ = P(X | U_1, \dots, U_m)$$



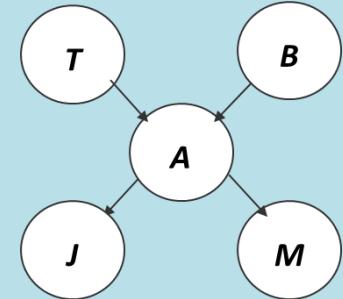
### Markov Blanket

Each node is conditionally independent of all other nodes given its Markov blanket (Parents + children + children's parents)

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 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  $p$  for  $X_i = \text{true}$
- 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 car thief network?
  - $1 + 1 + 4 + 2 + 2 = 10$  numbers (vs.  $2^5 - 1 = 31$ )



Why do we need only 31 and not 32 nodes for the network?

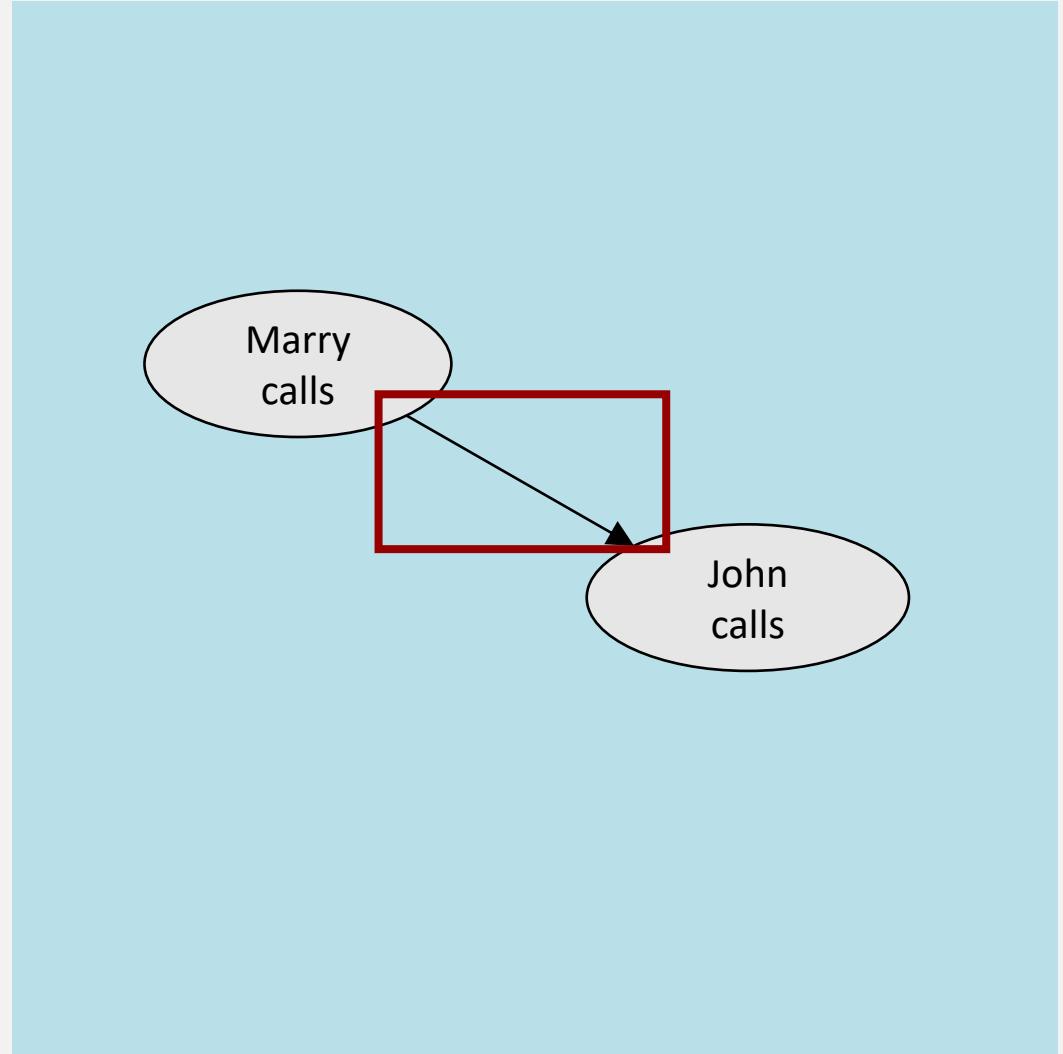
## 8.2 Constructing Bayesian Networks

- Identify the direct relationships of your network
  - $P(M | J, A, T, B) = P(M | A)$
- Choose an ordering of variables  $X_1, \dots, X_n$
- While you have remaining variables
  - add  $X_i$  to the network
  - select parents from  $X_1, \dots, X_{i-1}$  such that
$$P(X_i | Parents(X_i)) = P(X_i | X_1, \dots, X_{i-1})$$
 and define the probability table

Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

## 8.2 Example 1 ► Step 1

- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack
- $P(J | M) = P(J)?$
- If Marry calls, it is more likely that John calls as well



Adapted from Russell, S., & Norvig, P. (2016)

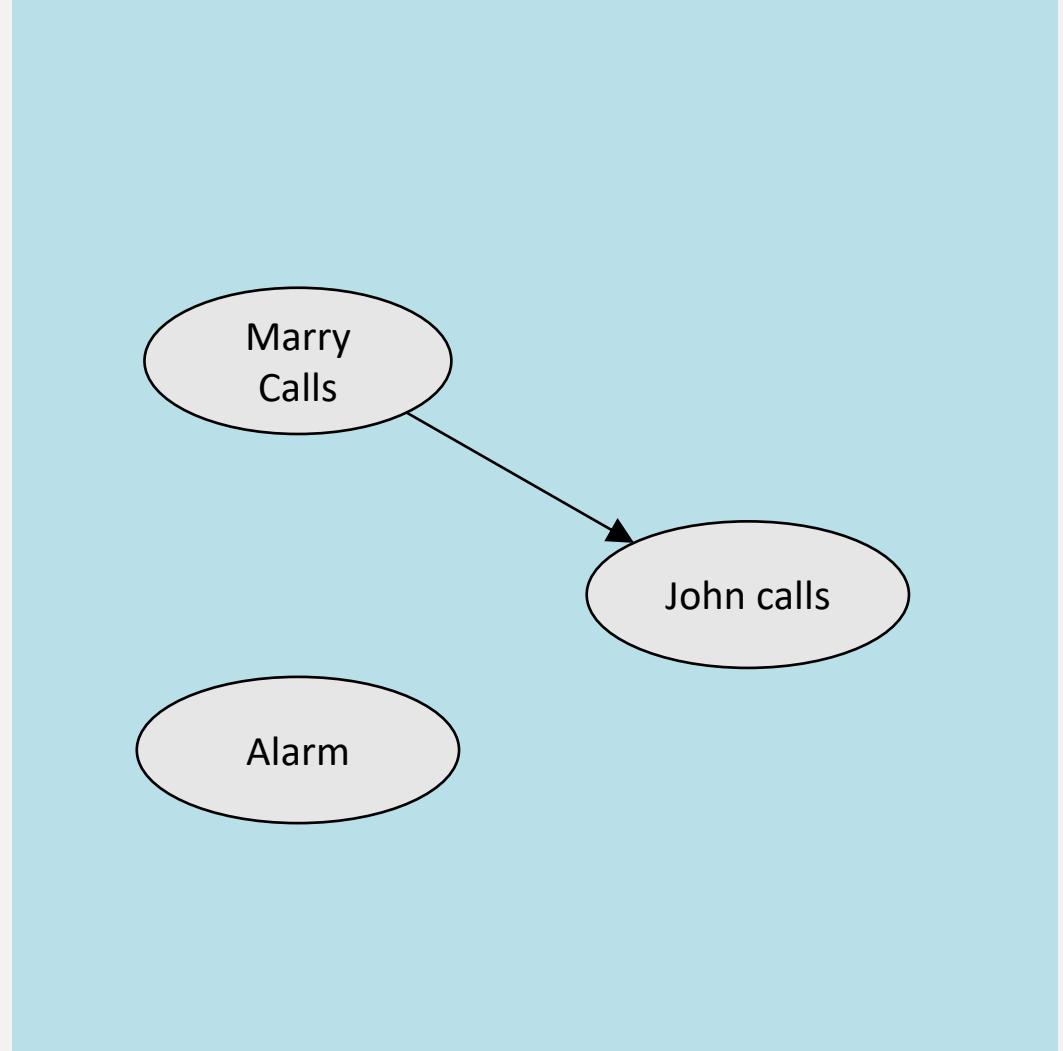
## 8.2 Example 1 ► Step 2

- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack

- $P(A | J, M) = P(A)?$

X

- If Marry and John call, the probability that the alarm has gone off is larger than if they don't call
- Node A needs parents J or M



Adapted from Russell, S., & Norvig, P. (2016)

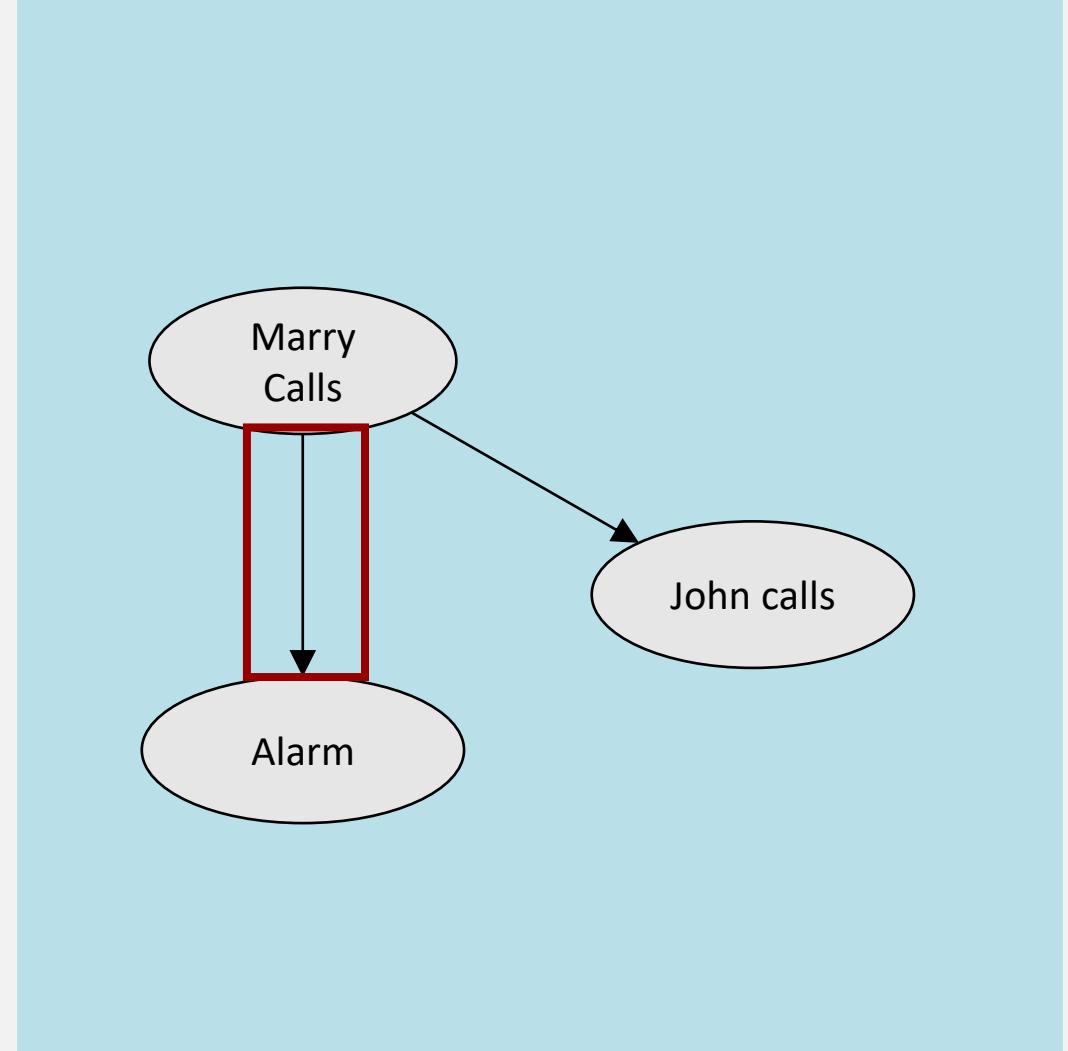
## 8.2 Example 1 ► Step 3

- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack

- $P(A | J, M) = P(A | J)?$

X

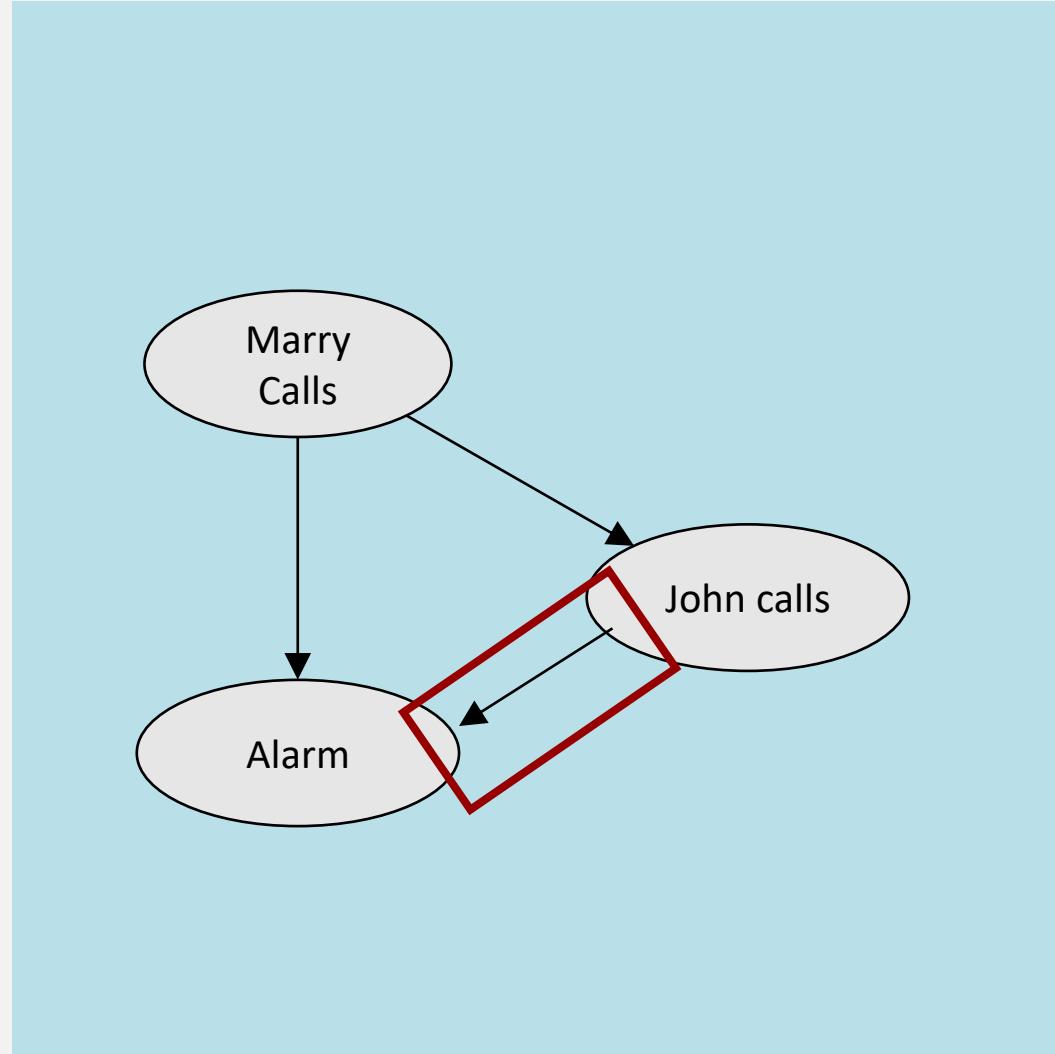
- If Marry and John call, the probability that the alarm has gone off is larger than if only John calls



Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Example 1 ► Step 4

- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack
- $P(A | J, M) = P(A | M)$ : 
- If Marry and John call, the probability that the alarm has gone off is larger than if only Marry calls



Adapted from Russell, S., & Norvig, P. (2016)

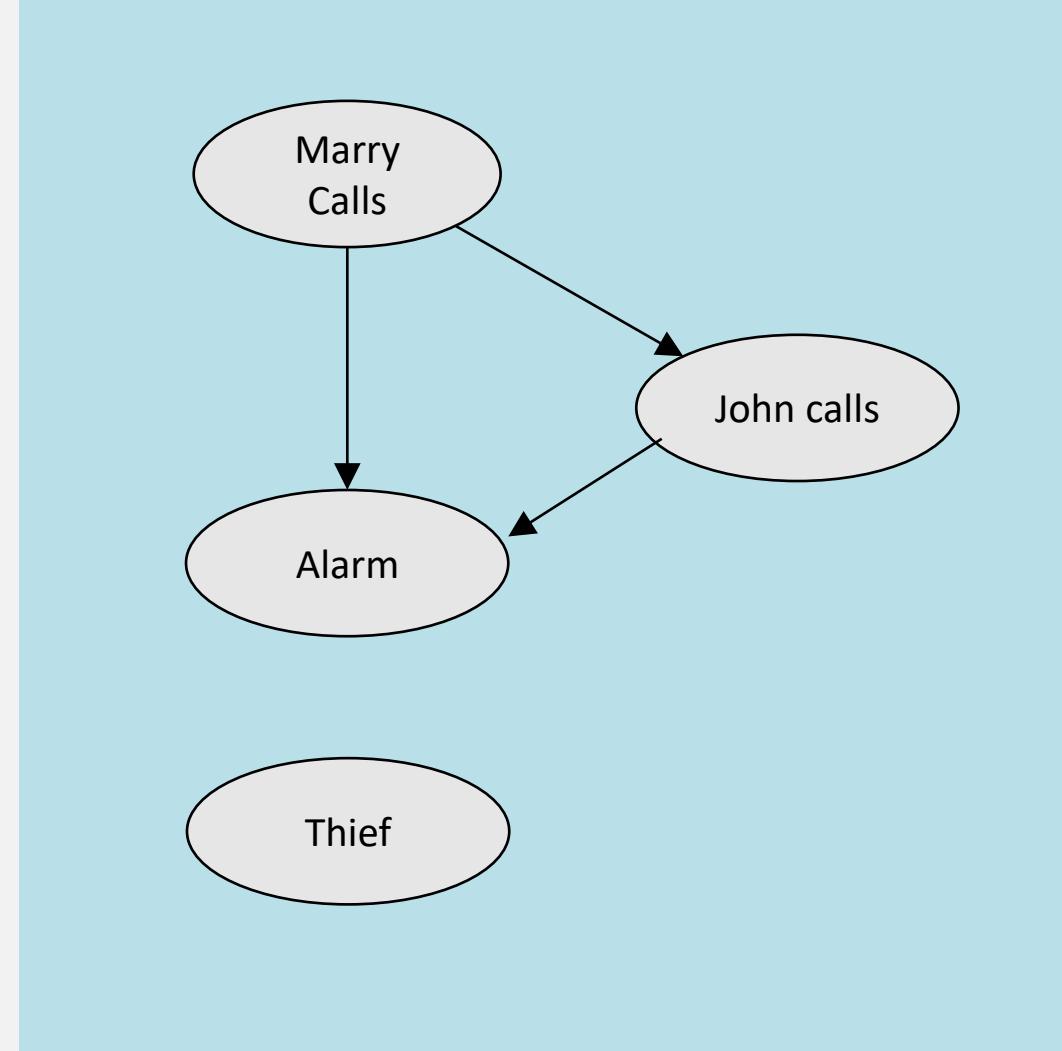
## 8.2 Example 1 ► Step 5

- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack

- $P(T | A, J, M) = P(T)?$

X

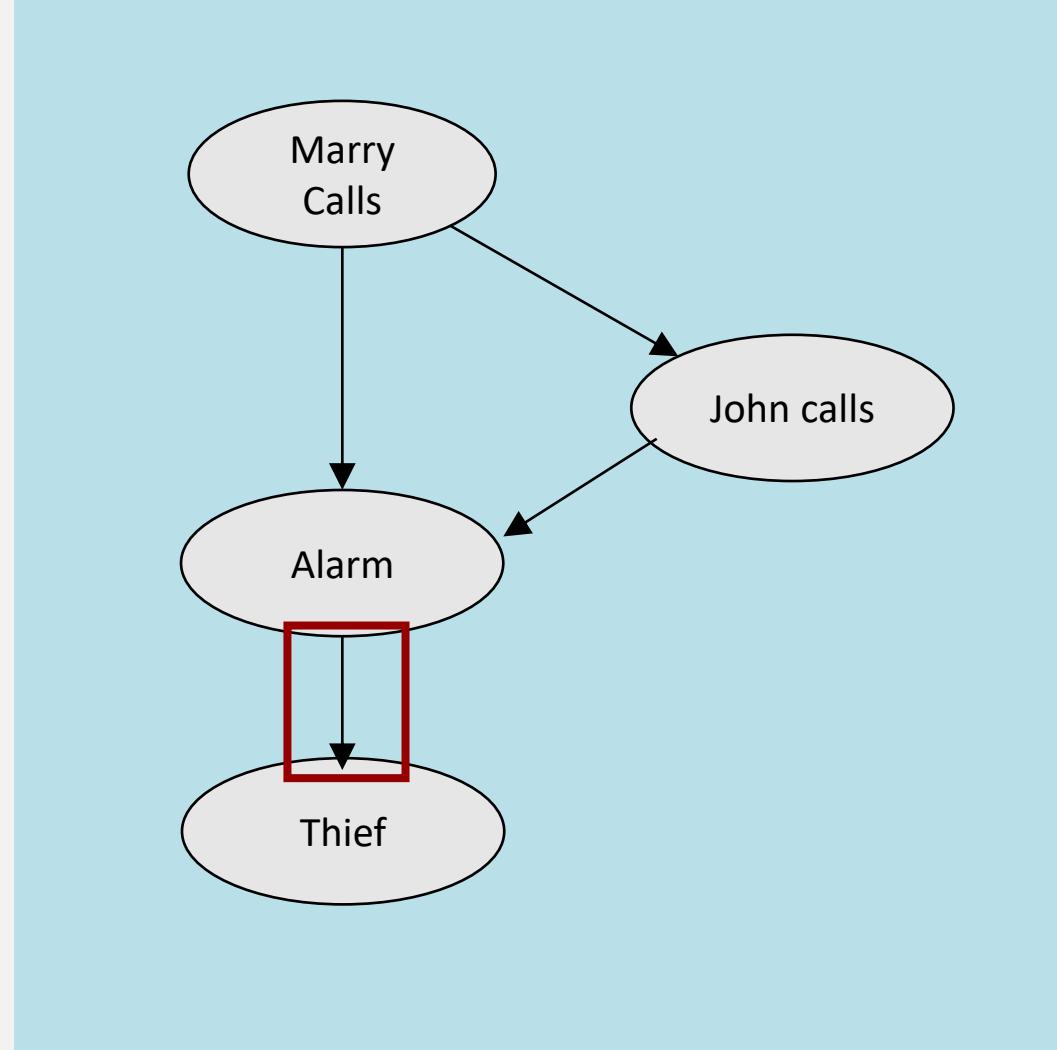
- Knowing whether Mary or John called and whether the alarm went off influences my knowledge about whether the car has been stolen
- Node T needs parents A, J, M



Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Example 1 ► Step 6

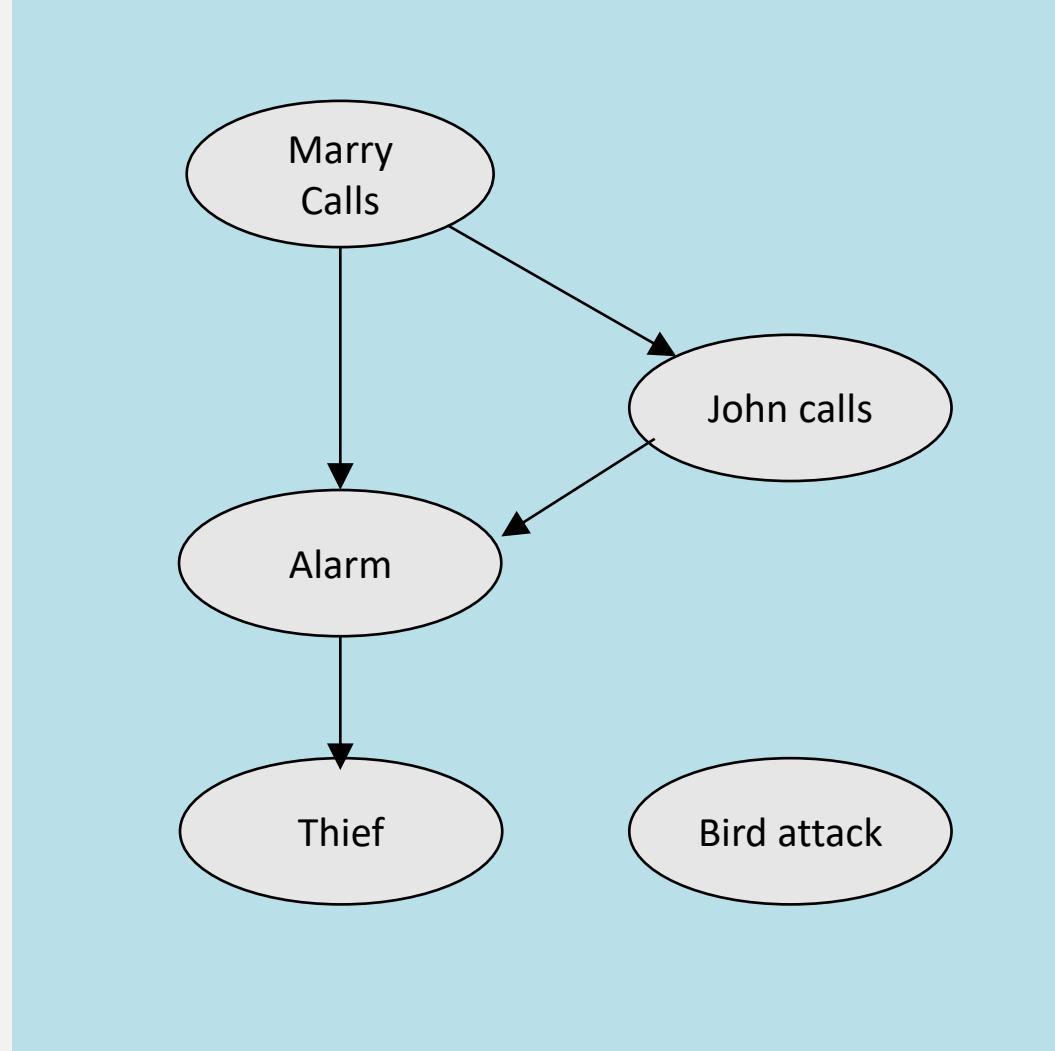
- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack
- $P(T | A, J, M) = P(T | A)?$  
- If I know that the alarm has gone off, knowing that John or Mary have called does not add to my knowledge of whether there has been a thief or not
- Thus, no edges from M and J, only from T



Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Example 1 ► Step 7

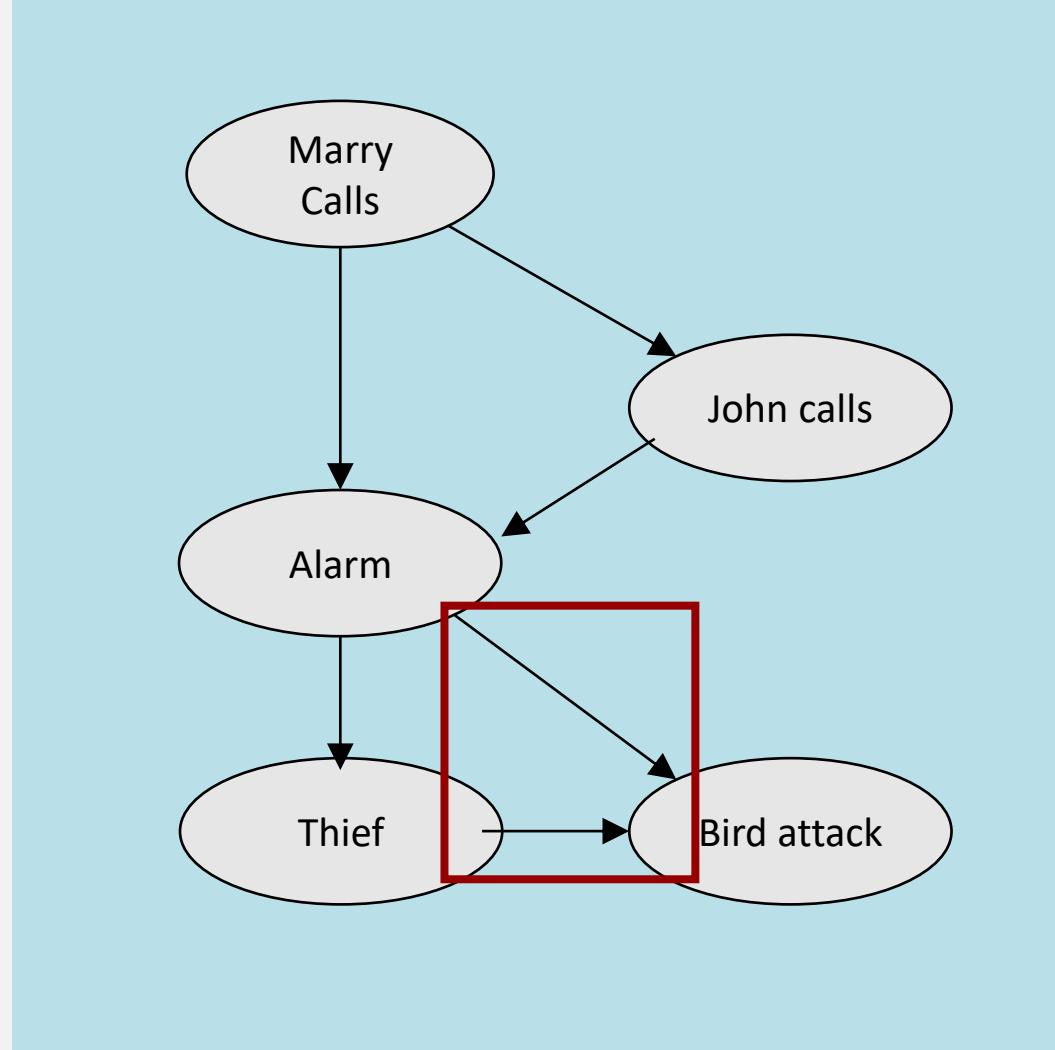
- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack
- $P(B | T, A, J, M) = P(B | A)?$  X
- Knowing whether there has been an alarm does not suffice to determine the probability of a bird shitting on my car, we have to know if my car is stolen or not



Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Example 1 ► Step 8

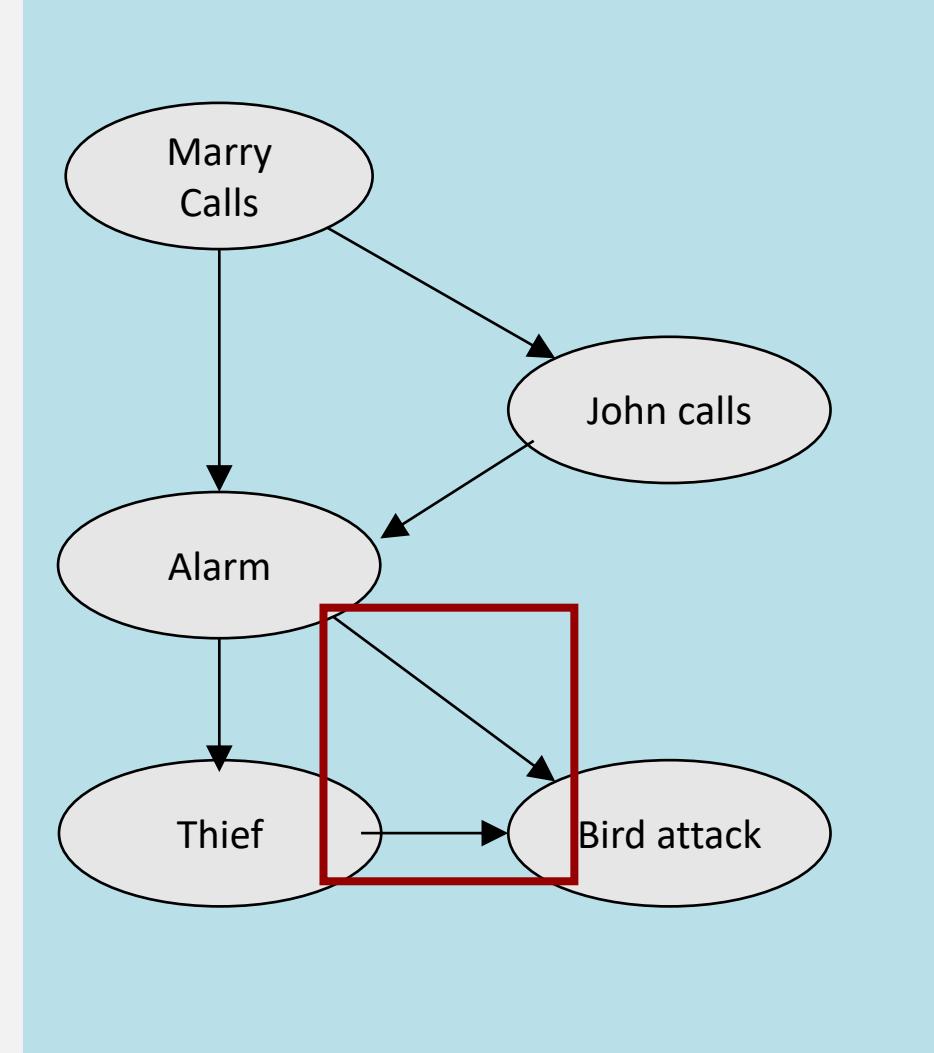
- Suppose we choose the ordering MarryCalls, JohnCalls, Alarm, Thief, BirdAttack
- $P(B | T, A, J, M) = P(B | A)?$
- $P(B | T, A, J, M) = P(B | A, T)?$
- Knowing whether there has been an alarm and whether my car has been stolen, no other factors will determine our knowledge about whether there has been a bird with digestive problems



Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Example 1 ► Discussion

- Deciding conditional independence is hard in noncausal directions
  - The causal direction seems much more natural
- Network is less compact:  $1 + 2 + 4 + 2 + 4 = 13$  numbers needed
- Worst possible ordering (fully connected network): Marry calls, John calls, Birth attack, Thief, Alarm



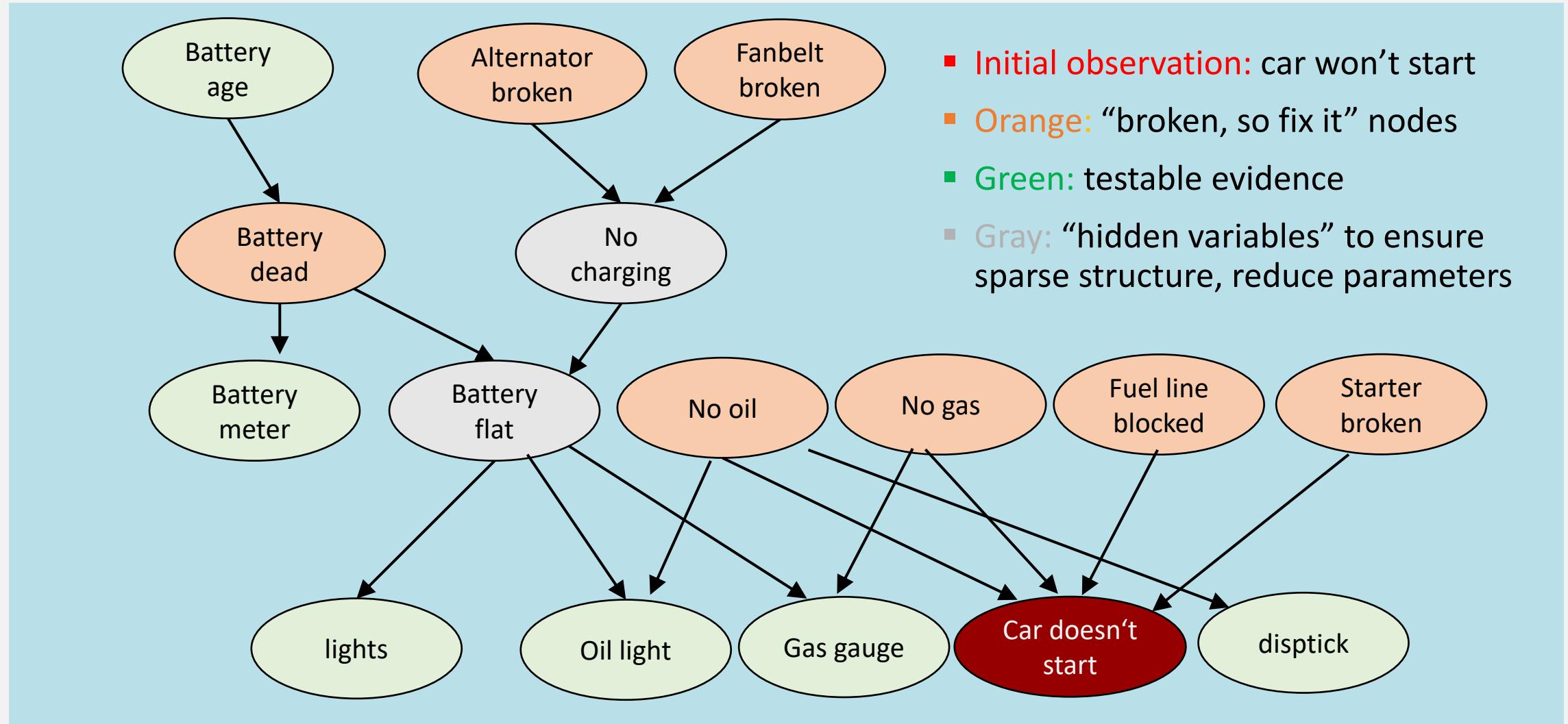
Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 Bayesian Networks and Continuous Variables

- Problems:
  - Conditional probability tables require discrete values
  - Many real-live scenarios have continuous variables
- Solution:
  - Discretization (see  “Chapter 4 – Data and Feature Engineering”)
  - Define probability density functions (require specific methods for combining discrete and continuous variables)

Adapted from Russell, S., & Norvig, P. (2016)

## 8.2 A more Realistic Car Diagnosis Network



Adapted from Russell, S., & Norvig, P. (2016); Frochte, J. (2020);

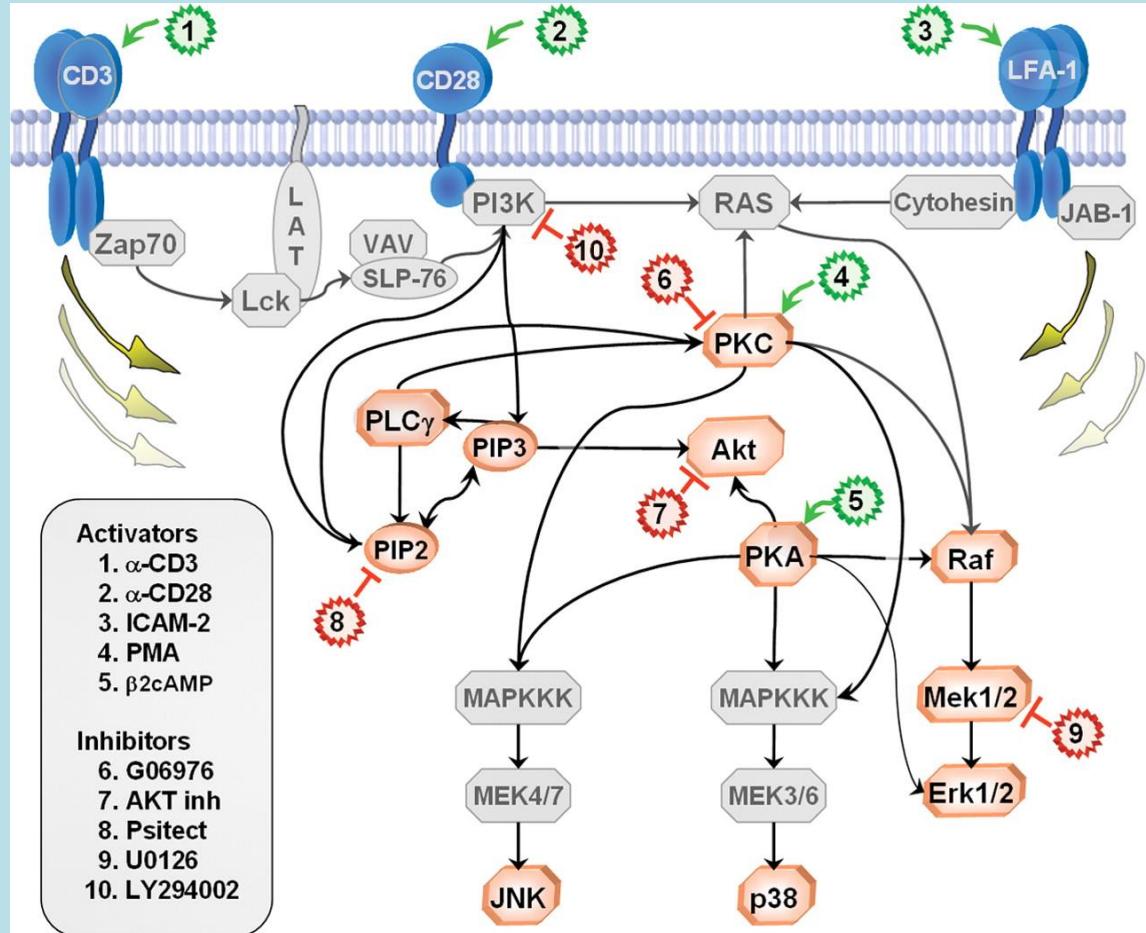
## 8.2 Application in Bioinformatics

- Probabilistic Modelling for the automated derivation of causal influences in cellular signaling network

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

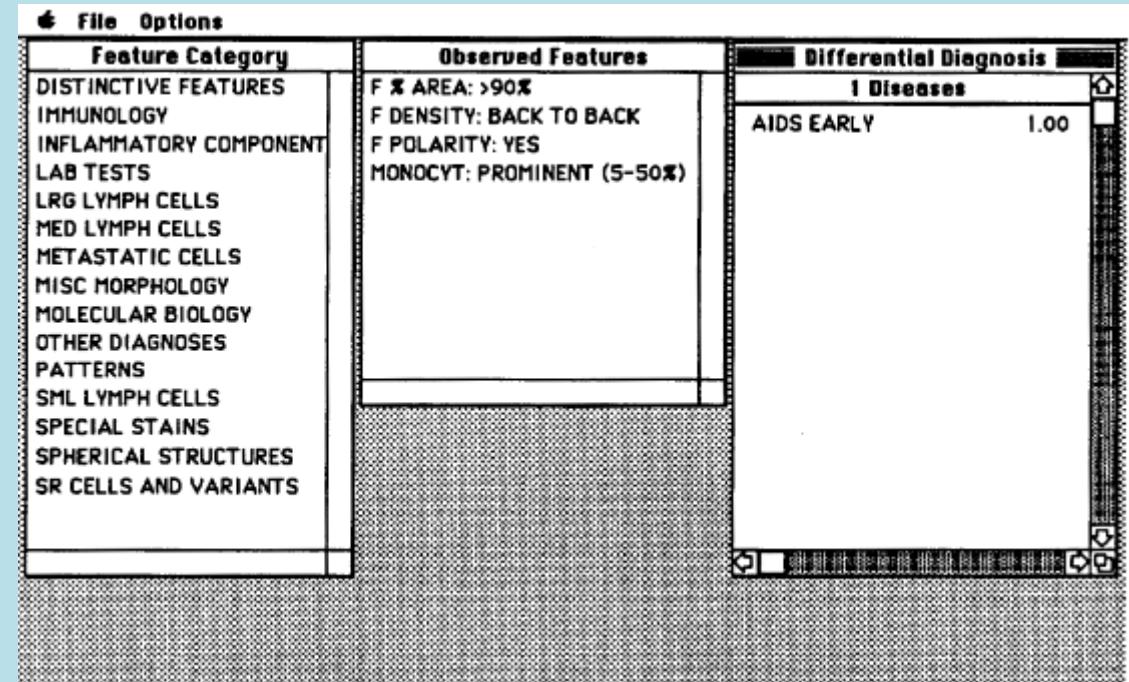
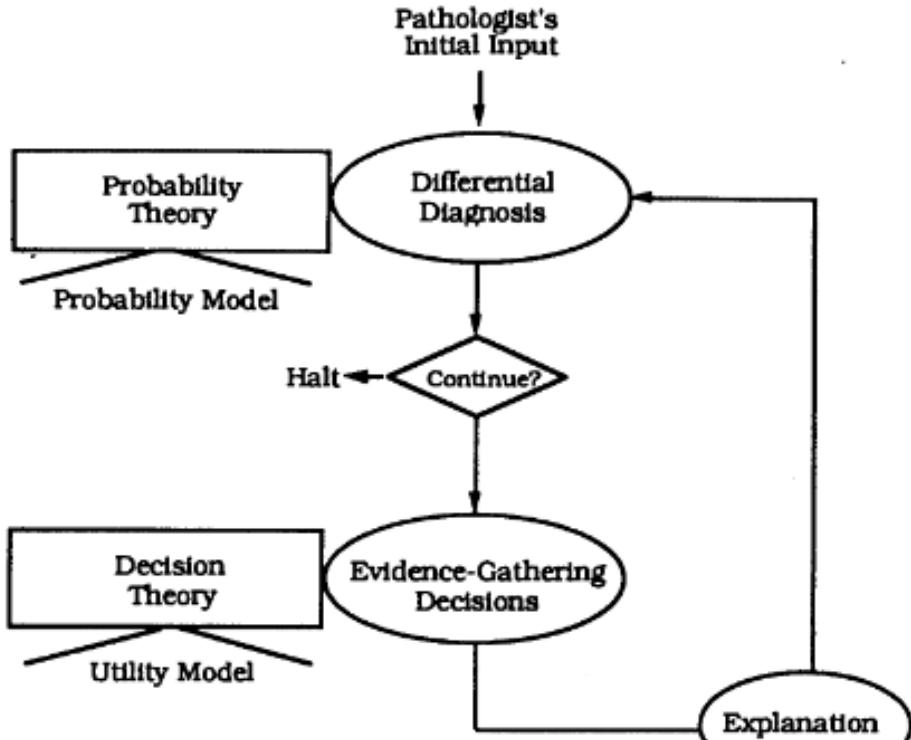
Karen Sachs,<sup>1\*</sup> Omar Perez,<sup>2\*</sup> Dana Pe'er,<sup>3\*</sup>  
Douglas A. Lauffenburger,<sup>1†</sup> Garry P. Nolan<sup>2†</sup>

Machine learning was applied for the automated derivation of causal influences in cellular signaling networks. This derivation relied on the simultaneous measurement of multiple phosphorylated protein and phospholipid components in thousands of individual primary human immune system cells. Perturbing these cells with molecular interventions drove the ordering of connections between pathway components, wherein Bayesian network computational methods automatically elucidated most of the traditionally reported signaling relationships and predicted novel interpathway network causalities, which we verified experimentally. Reconstruction of network models from physiologically relevant primary single cells might be applied to understanding native-state tissue signaling biology, complex drug actions, and dysfunctional signaling in diseased cells.



Adapted from Sachs K et al. (2005) | Image Source: Sachs K et al. (2005)

## 8.2 Expert System and Probabilistic Modelling



- Normative Expert Systems: Pathfinder IV System
- Research project (Stanford & Sth. California), expert systems based on decision theory for medical knowledge reasoning

Adapted from Heckerman, D. E. et al. (1992) | Image Source: Heckerman, D. E. et al. (1992)

# Your turn!

### Task

Given a bag of three biased coins  $a$ ,  $b$ , and  $c$  with probabilities of coming up heads of 20%, 60%, and 80%, respectively. One coin is drawn randomly from the bag (with equal likelihood of drawing each of the three coins), and then the coin is flipped three times to generate the outcomes  $X_1$ ,  $X_2$ , and  $X_3$ .

- Draw the Bayesian network corresponding to this setup and define the necessary CPTs.
- Calculate which coin was most likely to have been drawn from the bag if the observed flips come out heads twice and tails once.

## 8

### Probabilistic Reasoning and Modelling

8.1

From Uncertainty to Probability

8.2

Probabilistic Reasoning

8.3

Probabilistic Reasoning over Time

8.4

Decision Theory and Decision-Making

8.5

Game Theory and Sequential Decision-Making

Lectorial 6: Intelligent Agents in Action

► **What you will learn:**

- Concepts of statistics and probability theory to model agents that can act under uncertainty
- Foundations of Markov theory, Bayes theory and sequence analysis for to reason under uncertainty according to the laws of probability theory
- How to analyze and build optimal agent decision-making that can handle uncertainty

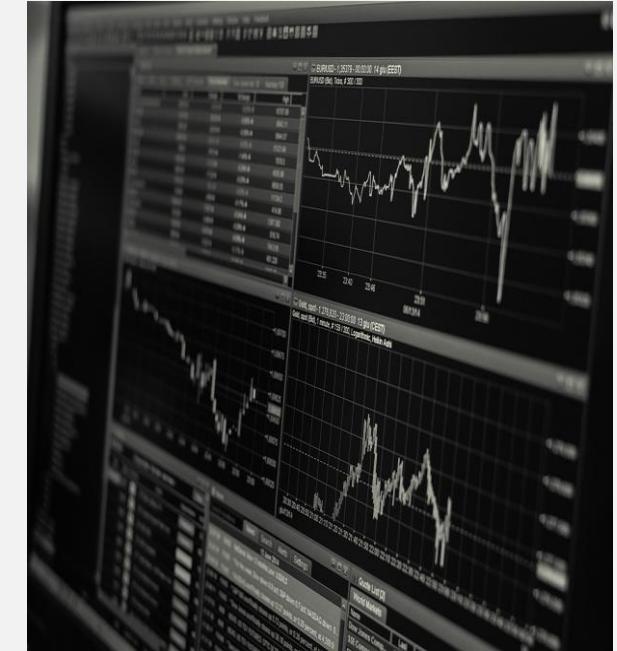


Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

► **Duration:**

- 215 min + 90 min (Lectorial)

► **Relevant for Exam:**

- 8.1 – 8.4

## 8.3 Example: Electric Vehicle Market in 2019

- You startet your career as an AI-Specialist at the Porsche AG. At the moment we assume the following three EVs in the European market:



- Let us suppose that your newest electric car, the Taycan controls now 20 % of the EV market. The competitors Model S controls 30 %, and the Renault Zoe 50%.
- The sales manager (best friend from your studytme at university) ask you to analyze the impact of a new marketing programm on the leasing market of EVs.

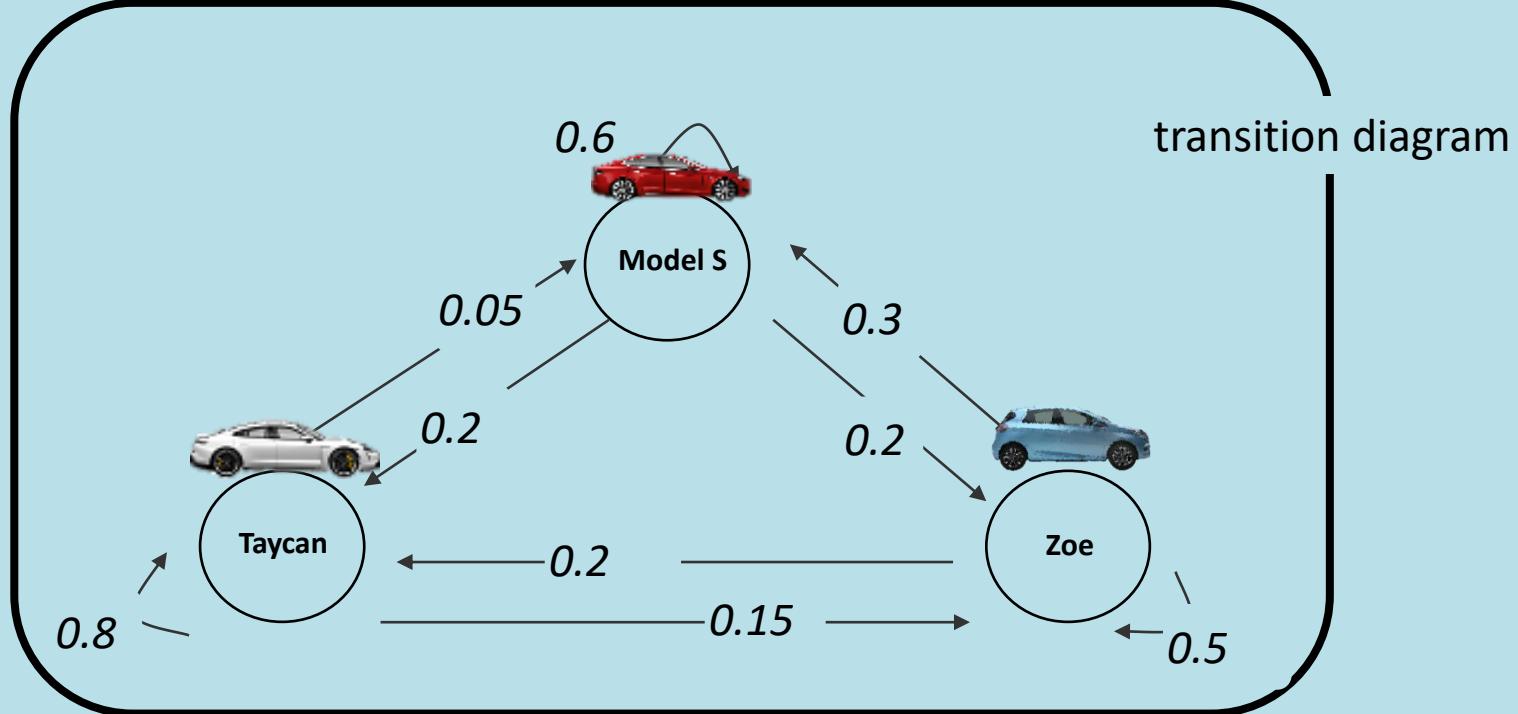
Suppose they find evidence that for the next buy:

- Someone driving Taycan will be so impressed that he continues to drive Taycan with 0.8 probability. The others will switch to Tesla with 0.05 and to the Renault Zoe with 0.15 probability.
- Someone driving Tesla will begin to drive Taycan with 0.2 probability. The others will continue to drive Tesla with 0.6 probability or start to drive Zoe with 0.2 probability.
- Someone driving Zoe will like the marketing campaign with 0.2 probability, so he will also start to drive Taycan. The others will switch to Tesla with 0.3, or stay to Renault with 0.5 probability.



How could you model these different dynamics in this case?

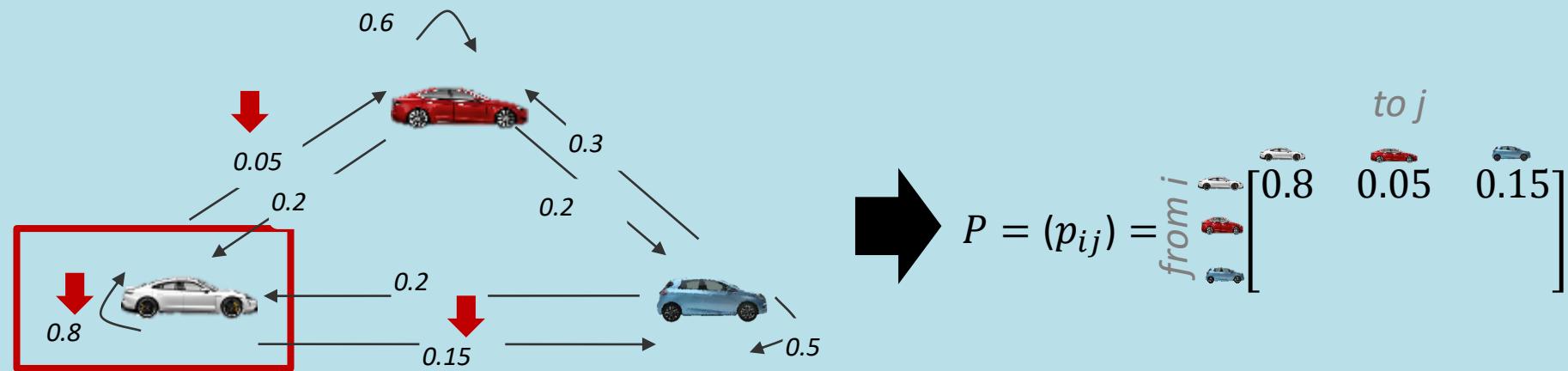
## 8.3 Markov Model of the Car Market



initial market segmentation with 3 states  $\pi_0 = [0.2, 0.3, 0.5]$

## 8.3 Transition probability matrix

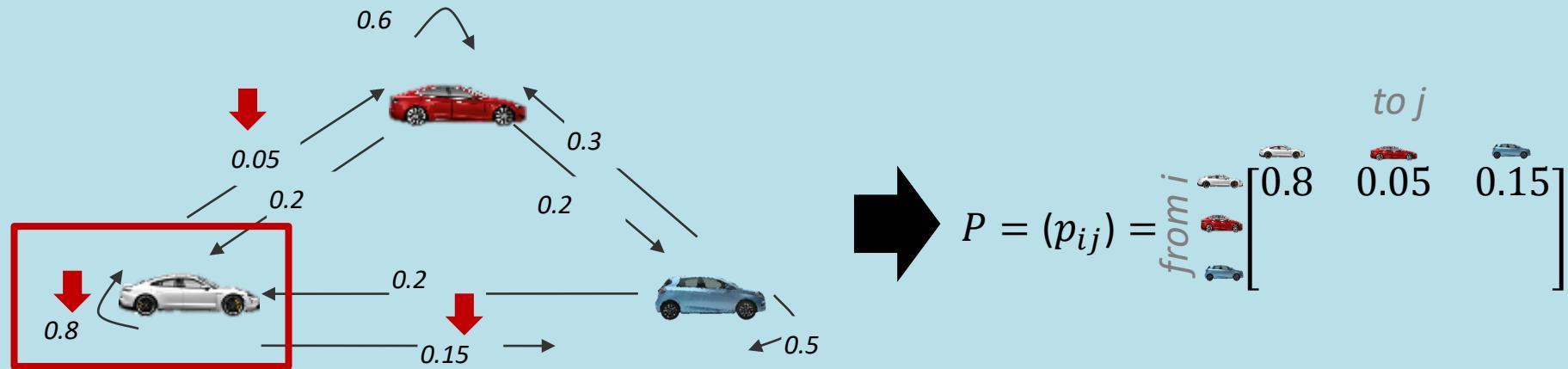
- Now we write this transition diagram in a more formal way, so that we can do some math with it...



- The row of the matrix indicates, that there is a 0.8 probability that you stay at Taycan ( $p_{TT} = 0.8$ ) and 0.05 that you switch to Tesla ( $p_{TM} = 0.05$ ) and 0.15 that you switch to Renault ( $p_{TR} = 0.15$ ).

## 8.3 Transition probability matrix

- And finally we get the transition probability matrix



Always check the row sums.  
They must be 1 (probabilities  
always sum up to 1)

## 8.3 Calculate market segmentation in 1 month

- The final transition probability matrix allows us, together with the initial state distribution, to calculate some probabilities...
- For instance: What is the probability that someone drives Taycan after 1 month or 1 timestep of the marketing campaign?

$$P = \begin{bmatrix} \text{white car} & \text{red car} & \text{blue car} \\ \text{white car} & 0.8 & 0.05 & 0.15 \\ \text{red car} & 0.2 & 0.6 & 0.2 \\ \text{blue car} & 0.2 & 0.3 & 0.5 \end{bmatrix}$$

$$\pi_0 = [\text{white car}, \text{red car}, \text{blue car}] = [0.2, 0.3, 0.5]$$

## 8.3 Calculate market segmentation in 1 month

$$P = \begin{bmatrix} \text{white car icon} & \text{red car icon} & \text{blue car icon} \\ 0.8 & 0.05 & 0.15 \\ \text{red car icon} & 0.2 & 0.6 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$
$$\pi_0 = [0.2, 0.3, 0.5]$$

$P(\text{Taycan}) = P(\text{drives Taycan and stays}) +$   
 $P(\text{drives Tesla and switches to}) +$   
 $P(\text{drives Renault Zoe and switches to})$

$$P(\text{Taycan}) = 0.2 \cdot 0.8 + 0.3 \cdot 0.2 + 0.5 \cdot 0.2 = 0.32$$

## 8.3 Classroom Task



Previous  
Exam Task!

# Your turn!

### Task

Please discuss with your neighbor:

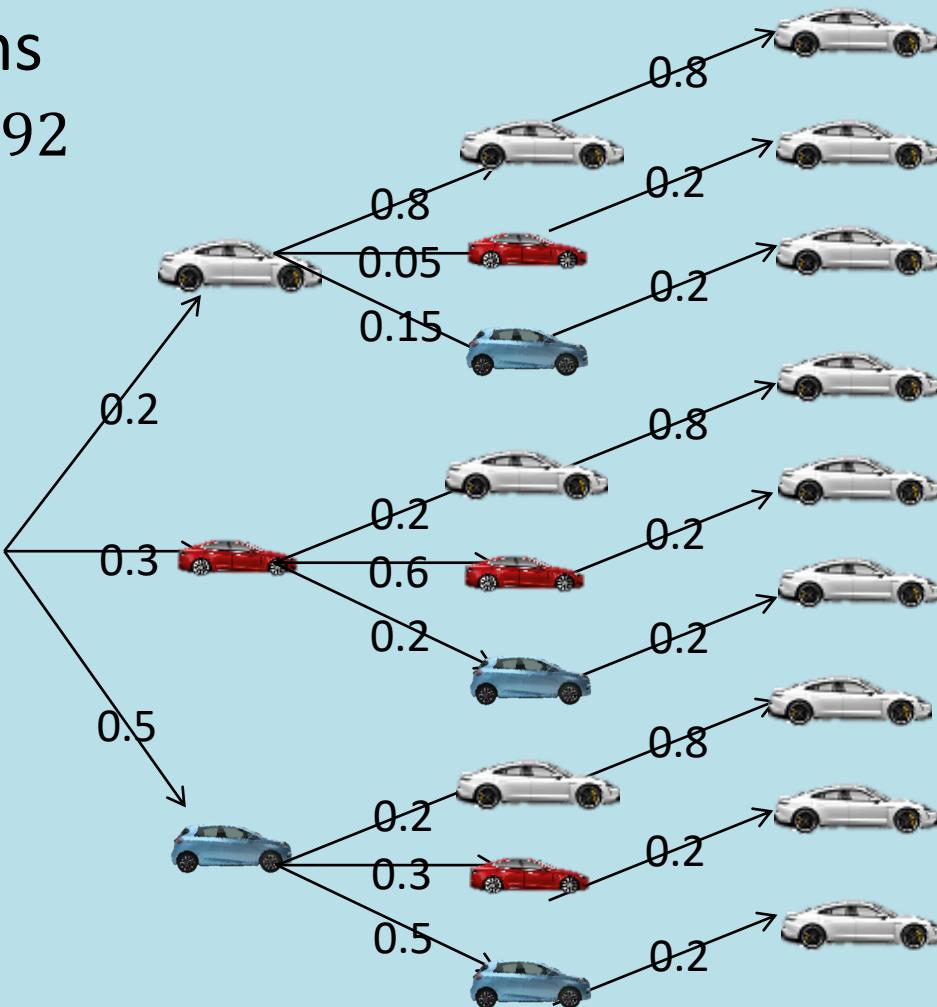
- How do you calculate the market segmentation?
- What is the market segmentation after 2 timesteps later?

$$P = \begin{bmatrix} \text{white car} & \text{red car} & \text{blue car} \\ \text{white car} & 0.8 & 0.05 & 0.15 \\ \text{red car} & 0.2 & 0.6 & 0.2 \\ \text{blue car} & 0.2 & 0.3 & 0.5 \end{bmatrix} \quad \pi_0 = [\text{white car}, \text{red car}, \text{blue car}] = [0.2, 0.3, 0.5]$$

## 8.3 Calculate Market Segmentation in t Months/Time Steps

- After 2 months

- $P(J)_2 = 0.392$



$$\begin{aligned} P(Taycan)_2 &= \\ &0.2 \cdot 0.8 \cdot 0.8 \\ &+ 0.2 \cdot 0.05 \cdot 0.2 \\ &+ 0.2 \cdot 0.15 \cdot 0.2 \\ &+ 0.3 \cdot 0.2 \cdot 0.8 \\ &+ 0.3 \cdot 0.6 \cdot 0.2 \\ &+ 0.3 \cdot 0.2 \cdot 0.2 \\ &+ 0.5 \cdot 0.2 \cdot 0.8 \\ &+ 0.5 \cdot 0.3 \cdot 0.2 \\ &+ 0.5 \cdot 0.5 \cdot 0.2 \end{aligned}$$

## 8.3 How can we do that better?

- If you take a look at our calculation, you can notice that we can rewrite it as:

$$\pi_1 = \pi_0 \cdot P$$
$$\pi_1 = [0.2 \quad 0.3 \quad 0.5] \cdot \begin{bmatrix} 0.8 & 0.05 & 0.15 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

$$P(\text{Taycan})_1 = 0.2 \cdot 0.8 + 0.3 \cdot 0.2 + 0.5 \cdot 0.2$$

$$P(\text{Taycan})_1 = 0.32$$



Remember:  
Matrix multiplication  
is rows by columns

$$\pi_1 = [P(\text{Taycan}), P(\text{Tesla}), P(\text{Zoe})] = [0.32, 0.34, 0.34]$$

## 8.3 State Probability Distribution

- And then we can write  $\pi_2$  as

$$\pi_2 = \pi_1 \cdot P = \pi_0 \cdot P \cdot P$$

$$\pi_2 = [\cancel{\pi_1}] \cdot [0.2 \quad 0.3 \quad 0.5] \cdot \begin{bmatrix} 0.8 & 0.05 & 0.15 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \cdot [0.8 \quad 0.05 \quad 0.15]^\top$$

- And

$$\pi_3 = \pi_2 \cdot P = \pi_1 \cdot P \cdot P = \pi_0 \cdot P \cdot P \cdot P$$

- Therefore



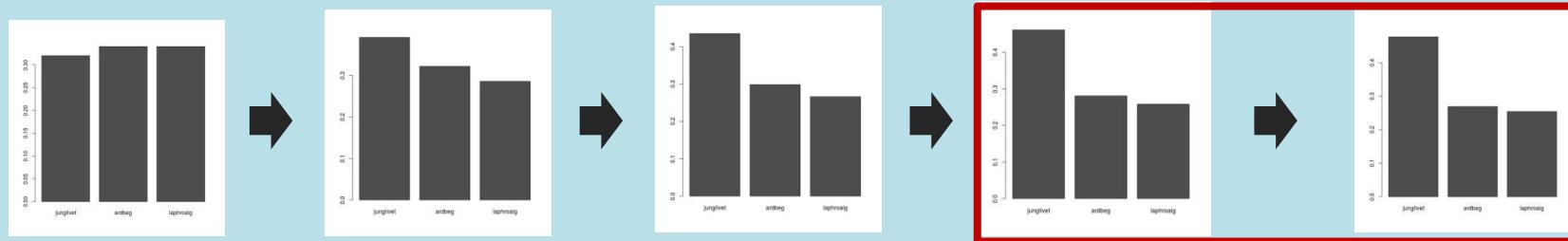
**State probability distribution:**

$$\pi_n = \pi_0 \cdot P^n$$

$$\pi_n = \pi_{n-1} \cdot P$$

## 8.3 Steady State

- If you calculate and plot the market segmentation



- It looks like that the segmentation is getting closer and closer to a stable distribution,  $\pi = [0.5, 0.25, 0.25]$

- In Markov chain terms, the matrix  $\pi$  is called **stationary** matrix
- And the system is in steady state, if the steady state is

$$\pi_\infty = \pi_\infty \cdot P$$

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Identify the Unique Stationary Matrix

- A regular Markov chain is a unique stationary matrix, that can be found by solving the equation:

$$\pi = \pi \cdot P$$

- Example:  $[u_1 \ u_2 \ u_3] = [u_1 \ u_2 \ u_3] \cdot \begin{bmatrix} 0.8 & 0.05 & 0.15 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$

$$u_1 \cdot 0.8 + u_2 \cdot 0.2 + u_3 \cdot 0.2 = u_1$$

$$u_1 \cdot 0.05 + u_2 \cdot 0.6 + u_3 \cdot 0.3 = u_2$$

$$u_1 \cdot 0.15 + u_2 \cdot 0.2 + u_3 \cdot 0.5 = u_3$$

$$s.t. \ u_i \geq 0; \sum u_i = 1$$

## 8.3 Identify the Unique Stationary Matrix II

$$\begin{aligned} u_1 \cdot 0.8 + u_2 \cdot 0.2 + u_3 \cdot 0.2 &= u_1 \\ u_1 \cdot 0.05 + u_2 \cdot 0.6 + u_3 \cdot 0.3 &= u_2 \\ u_1 \cdot 0.15 + u_2 \cdot 0.2 + u_3 \cdot 0.5 &= u_3 \end{aligned}$$

$$\begin{aligned} u_1 &= 2t \\ u_2 &= t \\ u_3 &= t \end{aligned}$$

Problem: dependent system

$$s.t. \quad u_i \geq 0$$

$$\sum u_i = 1 \quad \text{Solution: norm the probabilities}$$

## 8.3 Identify the Unique Stationary Matrix III

$$u_1 + u_2 + u_3 = 1$$
$$4t = 1 \rightarrow t = 0.25$$

$$u_1 = 2t$$

$$u_2 = t$$

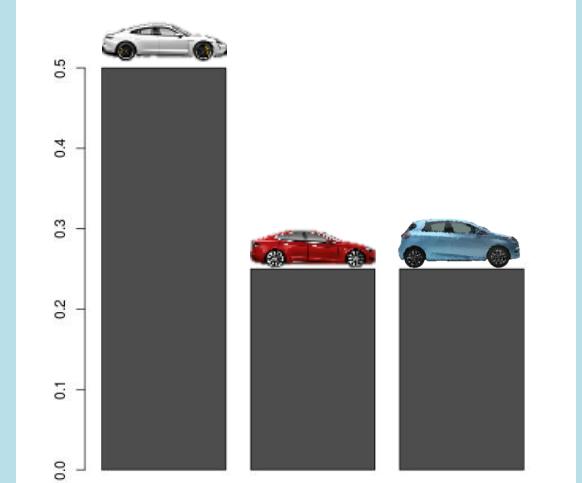
$$u_3 = t$$

$$u_1 = 0.5$$

$$u_2 = 0.25$$

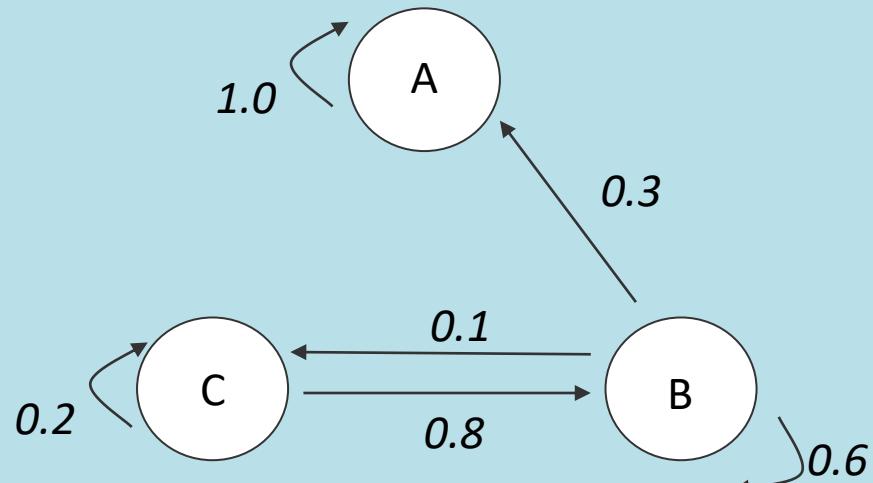
$$u_3 = 0.25$$

Barplot of stationary matrix



## 8.3 Markov Chain

► Represent data as consecutive decisions.



### Example

- User behaviour (user analytics), supply chain flow

- Random walk in a state space.
- A Markov chain must satisfy the Markov property, which says that if you want to predict where the chain will be at a future time, and if we know the present state, then the entire past history is irrelevant

- + Parameters can be easily calculated, speed of modelling, decision process can be represented graphically
- You have to model the states(need system knowledge), and the states must be observable, very simple method

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Definition: Markov Property

D

- Random walk in a state space  $S$ , which we will assume is finite, say  $n \in N_0$  and  $s_0, \dots, s_{n-1}, s_n, s_{n+1} \in S$ .
- We let  $X_t$  denote which element of  $S$  the walk is visiting at the time  $t$ .
- A Markov chain must satisfy the Markov property, which says that if you want to predict where the chain will be at a future time, and if we know the present state, then the entire past history is irrelevant

$$P(X_{n+1} = s_{n+1} | X_0 = s_0, X_1 = s_1, \dots, X_n = s_n) = P(X_{n+1} = s_{n+1} | X_n = s_n)$$

## 8.3 Definition: Markov Property

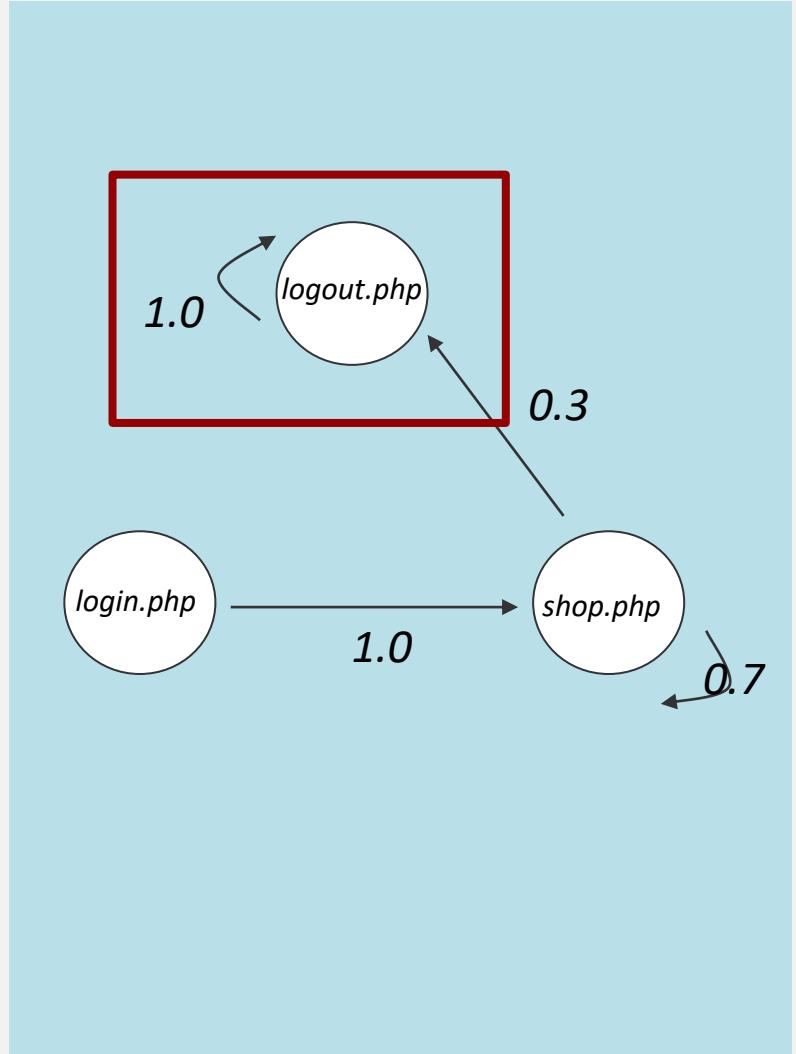
D

- Random walk in a state space  $S$ , which we will assume is finite, say  $n \in N_0$  and  $s_0, \dots, s_{n-1}, s_n, s_{n+1} \in S$ .
- We let  $X_t$  denote which element of  $S$  the walk is visiting at the time  $t$ .
- Mathematical way of saying:  
**„Today is the first day of the rest of your life“**

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Absorbing Markov Chains I

- A state in a Markov chain is called **absorbing**, if once the state is entered it is impossible to leave it
- In general absorbing markov chains can be treated like regular markov chains



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Absorbing Markov Chains II

- The presence of an absorbing state does not guarantee that the chain approach a stationary matrix

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \rightarrow \pi = [0.5 \ 0 \ 0.5] \text{ or } [0 \ 1 \ 0]$$

- To proof that, just check if the power of  $P$  starts oscillating
- A Markov chain is absorbing, if: there is at least one absorbing state and it is possible to go from each non-absorbing state to at least one absorbing state in a finite number of steps

## 8.3 Technical Definitions: Transition Matrix

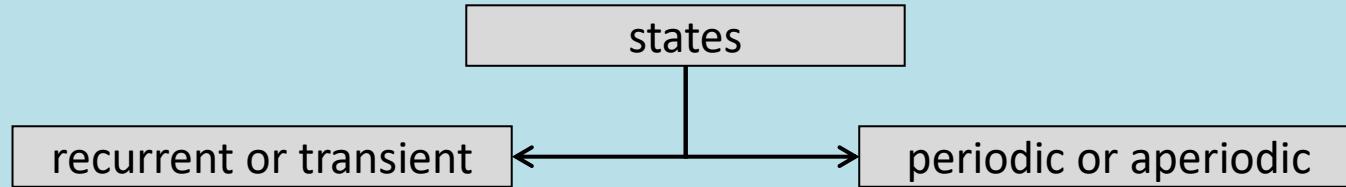
- Let the state space be  $I = \{1, 2, \dots, M\}$ . The transition matrix  $P$  is the  $M \times M$  matrix where element  $p_{ij}$  is the probability that the chain goes from state  $i$  to state  $j$  in one step

$$p_{ij} = P(X_{n+1} = s_j | X_n = s_i)$$

- To find the probability that the chain goes from state  $i$  to state  $j$  in exactly  $m$  steps, take the  $(i, j)$  element of  $P^m$

$$p_{ij}^{(m)} = P(X_{n+m} = s_j | X_n = s_i)$$

## 8.3 Technical Definitions: State Properties

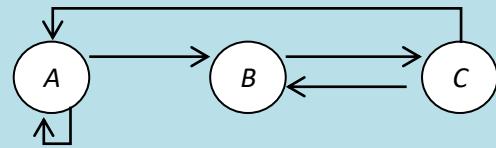


- A state is either **recurrent or transient**.
  - if you start at a recurrent state, then you will always return back to that state at some point in the future.
  - otherwise you are at a transient state. There is some positive probability that once you leave you will never return.
- A state is either **periodic or aperiodic**.
  - if you start at a periodic state of period  $k$ , then the greatest common divisor (GCD) of the possible numbers of steps it would take to return back is  $k > 1$ .
  - otherwise you are at an aperiodic state. The GCD of the possible numbers of steps it would take to return back is 1.

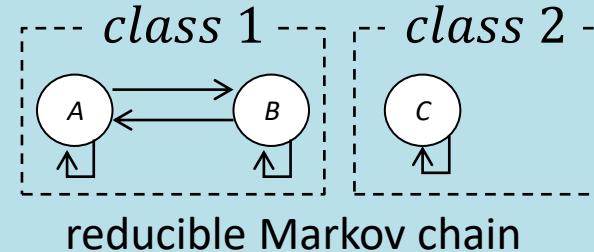
Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Technical Definitions: Chain Properties

- A chain is **irreducible** if you can get from anywhere to anywhere.
- If a chain (on a finite state space) is irreducible, then all of its states are recurrent.



irreducible Markov chain



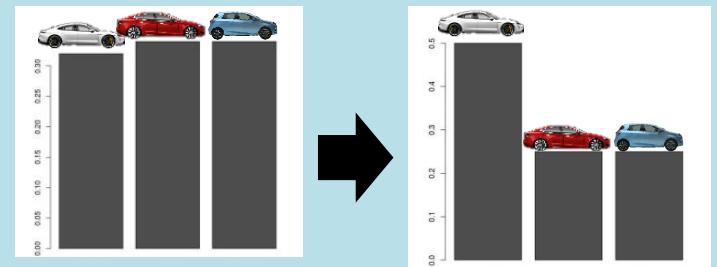
reducible Markov chain

- A chain is **periodic** if any of its states are periodic, and is **aperiodic** if none of its states are periodic. In an irreducible chain, all states have the same period.

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Technical Definitions: Stationary Distribution

- Let us say that the vector  $\pi$  be a probability mass function of the different states
- We call the the vector  $\pi$  a stationary distribution if  $\pi = \pi \cdot P$
- As a consequence if  $X_t$  has the stationary distribution  $\pi$ , then all future  $X_{t+1}, X_{t+2}, \dots$  also have the stationary distribution
- For irreducible, aperiodic chains, the stationary distribution  $\pi$  exists, is unique, and is the long-run probability of a chain being at state  $i$ .



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Classroom Task



Previous  
Exam Task!

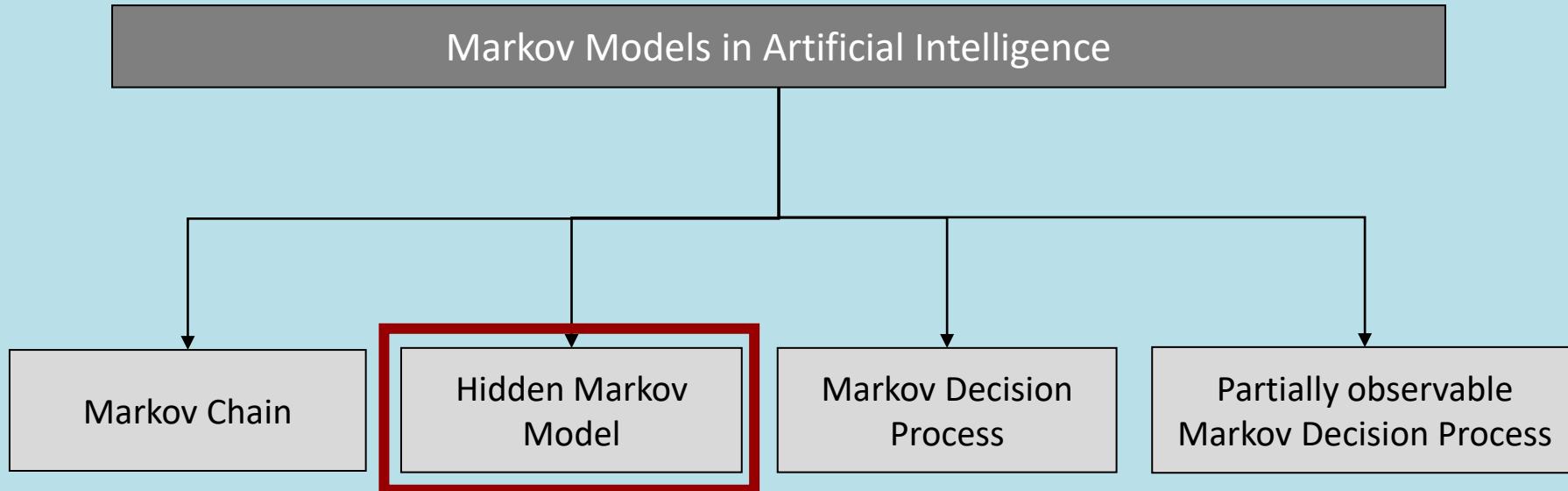
# Your turn!

### Task

Please discuss with your neighbor:

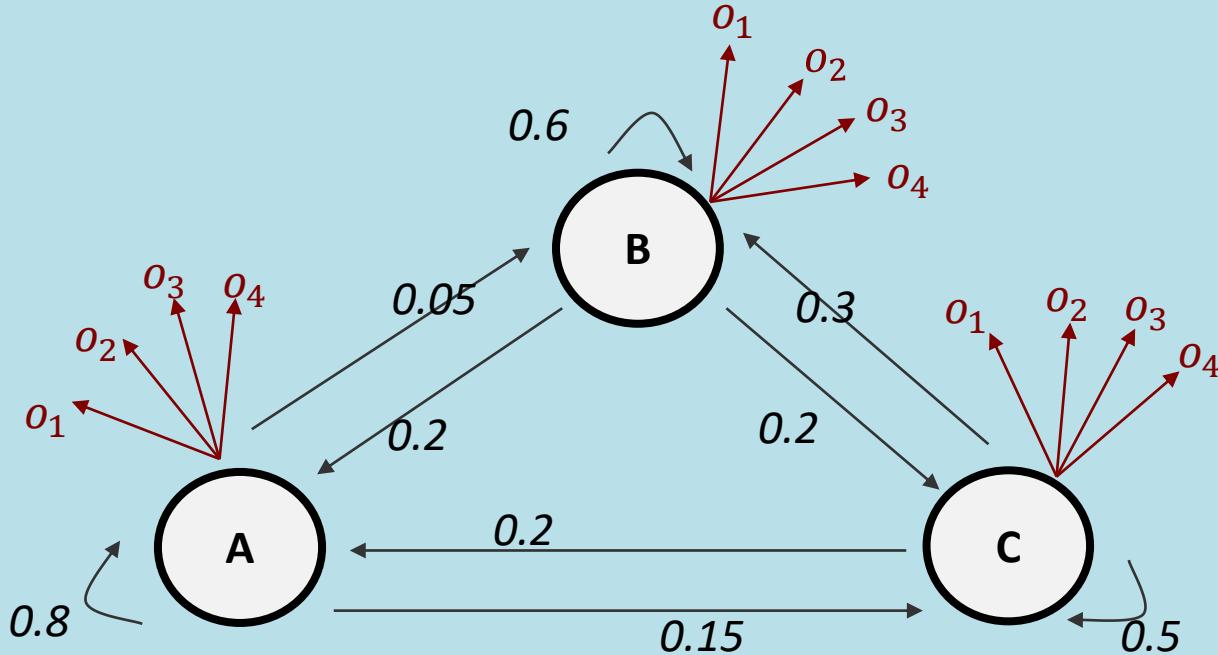
- How do you identify the percentage of consumer that use a product after 1, 2 or 3 timesteps?
- What is the initial state matrix?
- How do you compute the percentage of consumer that use a product in the long run?

## 8.3 Types of Markov Models



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

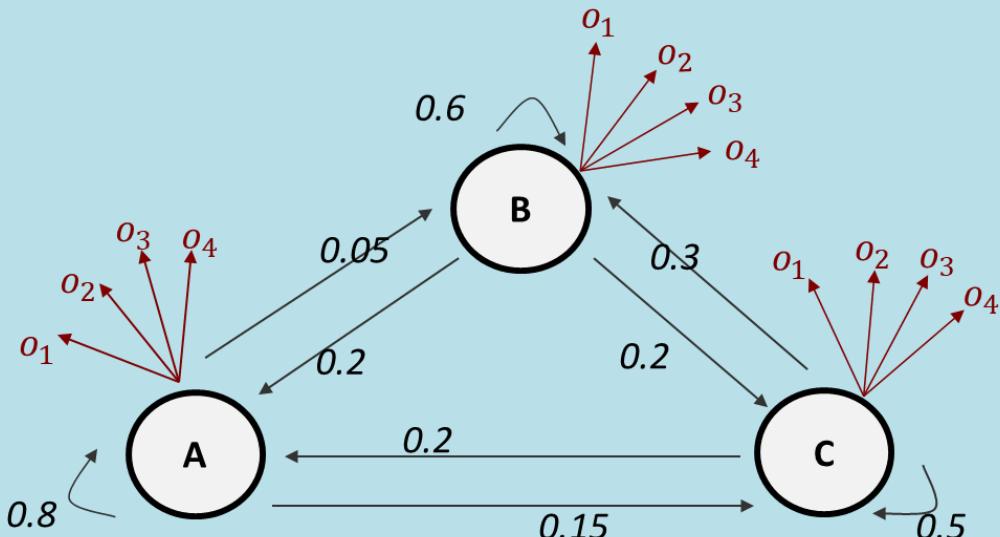
## 8.3 Conceptual Structure of a HMM



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Hidden Markov Models (HMM)

► Model an unobservable process that generates outcomes by observing these outcomes



### Example

- Bioinformatics, speech recognition, user analytics

- Observed or future states  $Y$  (e.g. purchase) are outcomes of an unobserved markov model (e.g. preferences)
- States space  $S$ , which we will assume is finite, say  $n \in N_0$  and  $s_0, \dots, s_{n-1}, s_n, s_{n+1} \in S$ , and state transition matrix  $P$  where element  $p_{ij}$  is the probability that the chain goes from state  $s_i$  to state  $s_j$  in one step  $t$
- Emission probability matrix  $B$  with probabilities  $b_{ij}$  and observations space  $O$ , which is finite, say  $n \in N_0$  and  $o_0, \dots, o_{n-1}, o_n, o_{n+1} \in O$

- + well studied understandable AI, algorithms learn fast, HMM are able to capture hidden states
- HMM can only model problems with a finite number of static distributions, time spent in a state is not captured explicitly

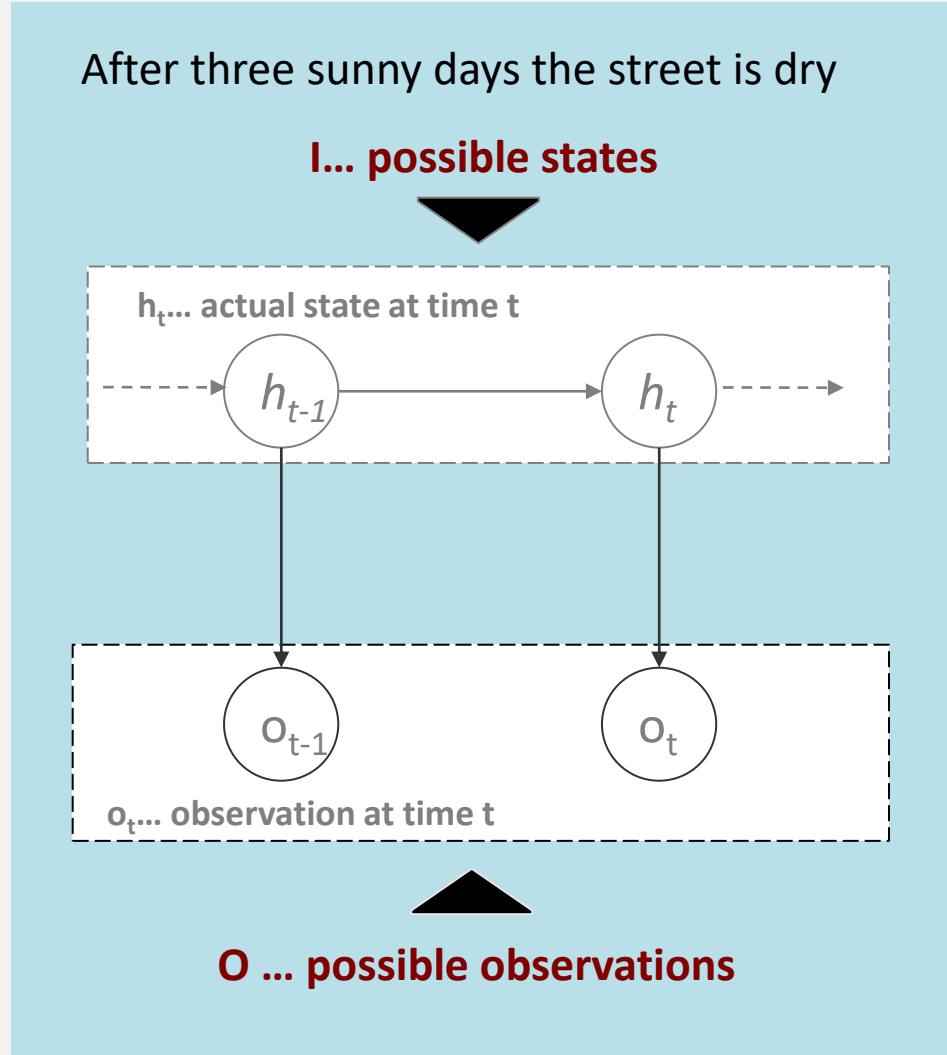
Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Example 2: Plan your National Park Roadtrip

- You and your girl/boyfriend plan to make a **spontaneous kayak tour** in the Everglades National Park .
- However **the weather is extremely changeable in spring**. So there are no accurate weather reports and **you have to rely on the ancient knowledge from your native grandmother**.
- She says, that **after three sunny days** the spring is coming and the changes are good for the kayak weekend. So you have to **check out the streets on the national parks webcam** (if they are dry, rather dry, rather wet, wet) to decide when to start your tour.



## 8.3 Use a Hidden Markov Model (HMM) for that!



## 8.3 State and Observation

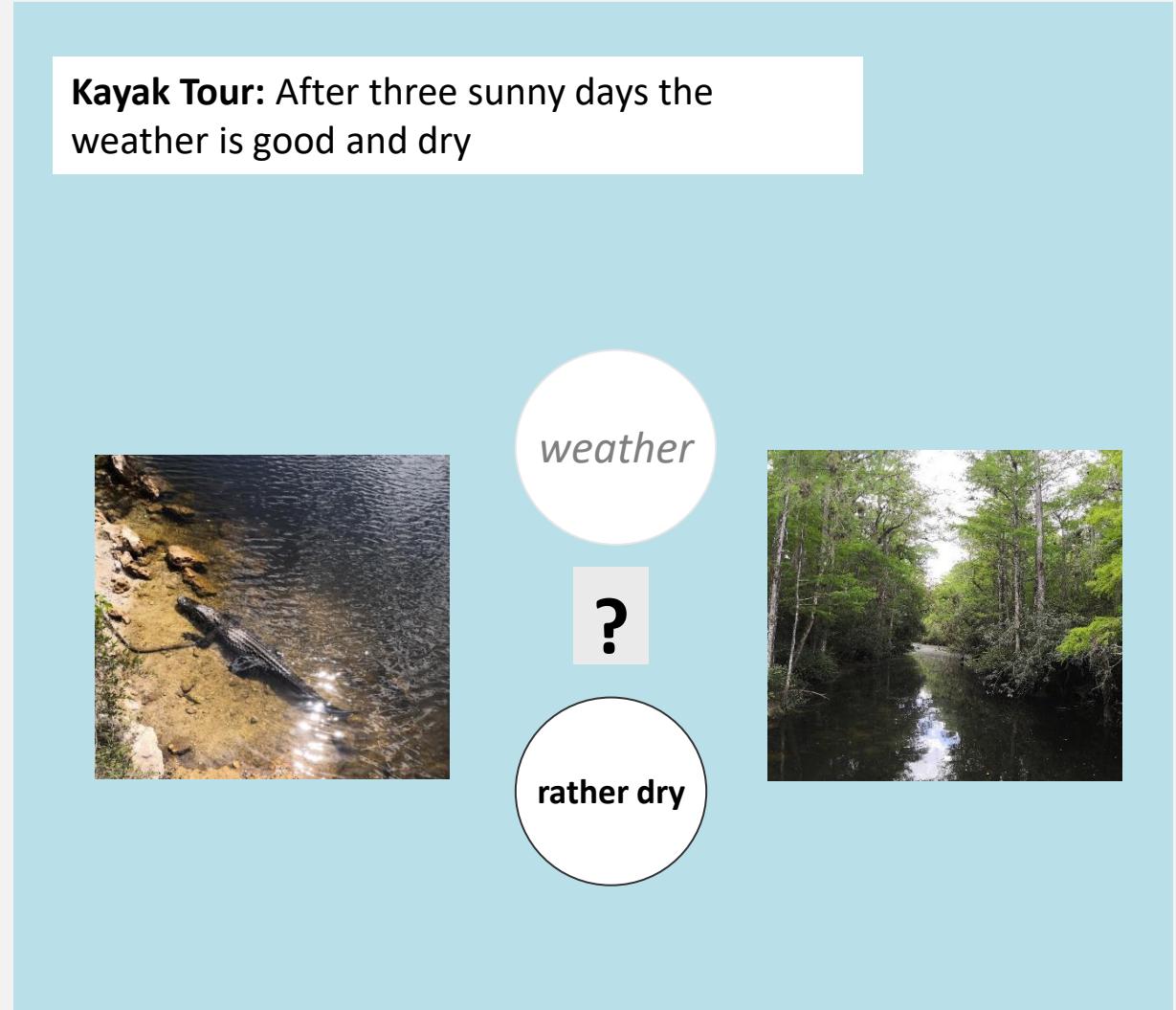
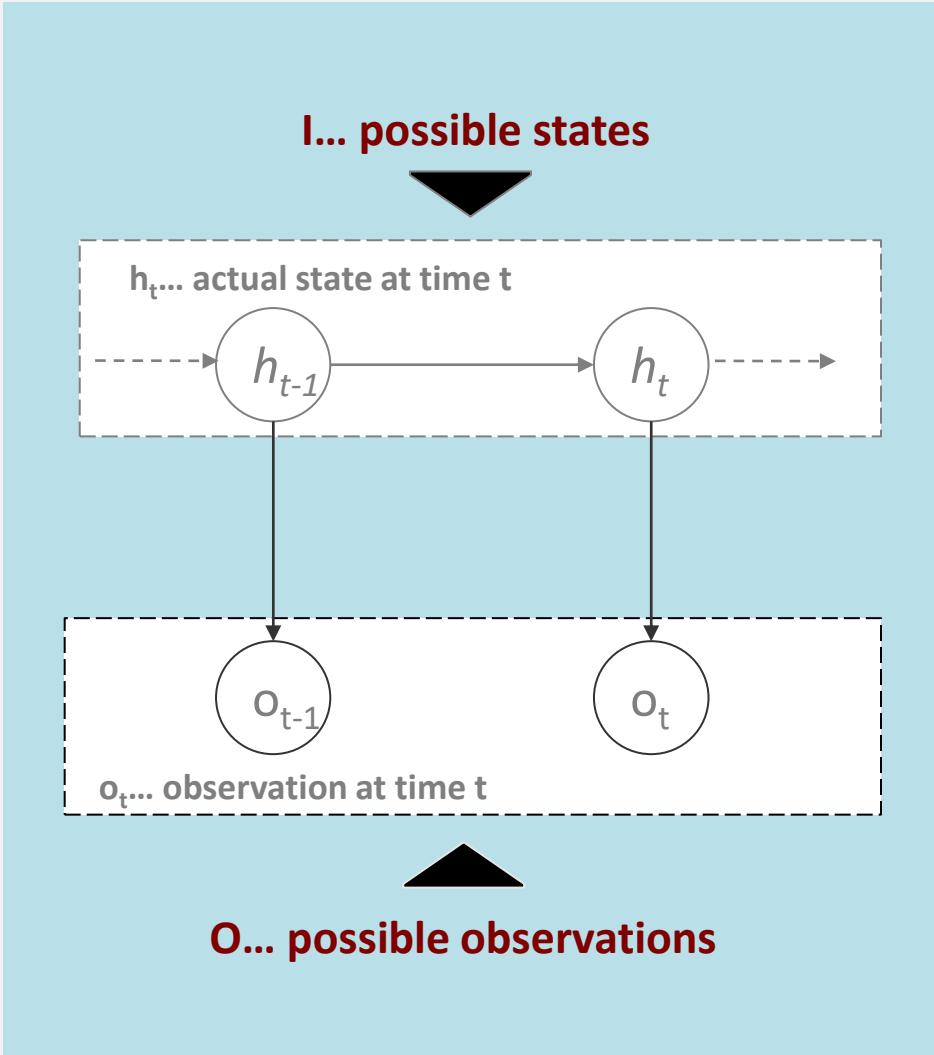
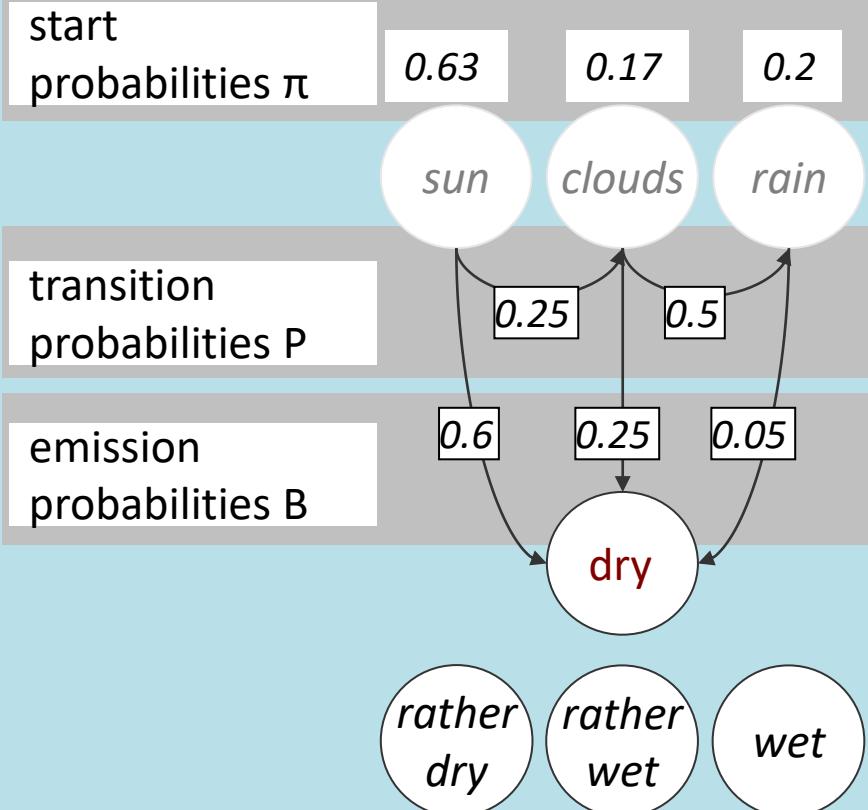


Image source: ↗ Pixabay (2019) / ↗ CCO

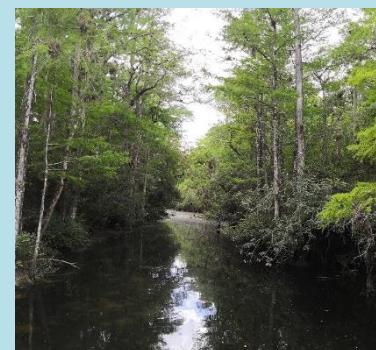
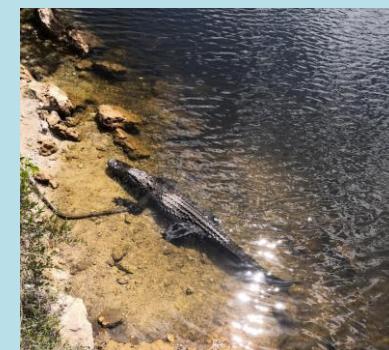
## 8.3 Parameters in HMM



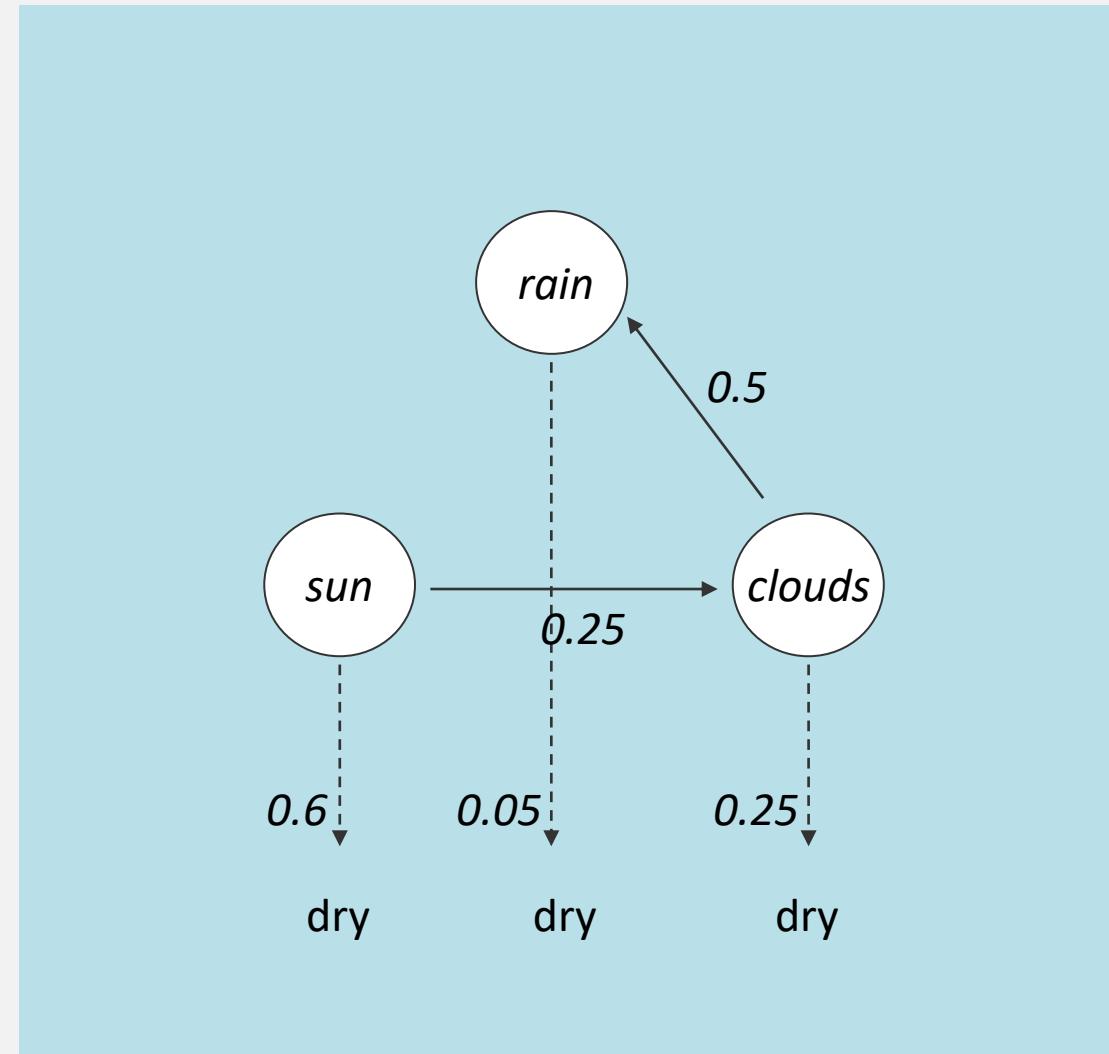
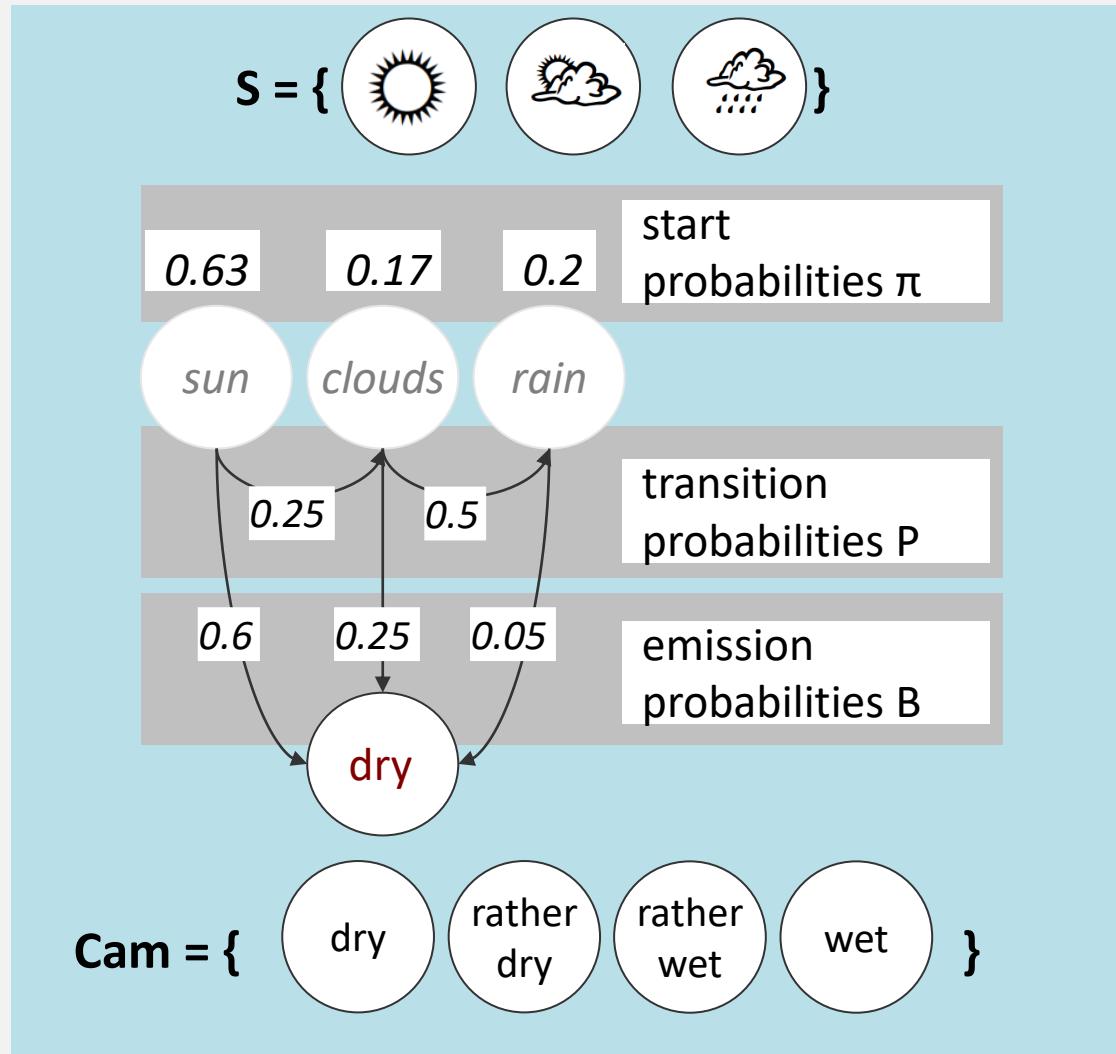
### You know

- which states exist
- how frequently they occur
- which observations may occur
- how probable it is that a sunny day is followed by a rainy day etc.
- how probable it is that the cloth is dry on a sunny day, on a rainy day, etc.

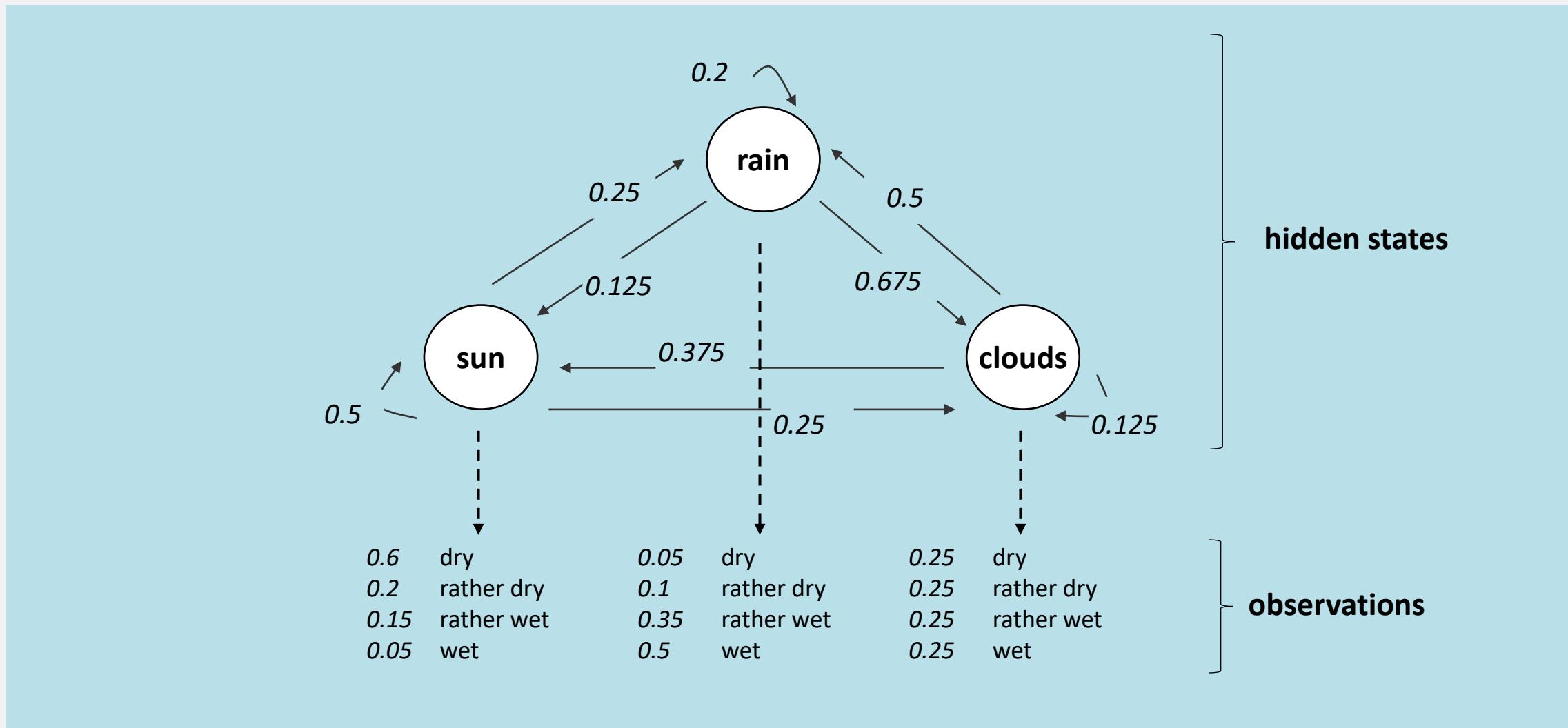
S  
π  
O  
P  
B



## 8.3 Markov-Model Completion



## 8.3 Modelling the Final Hidden Markov Process



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

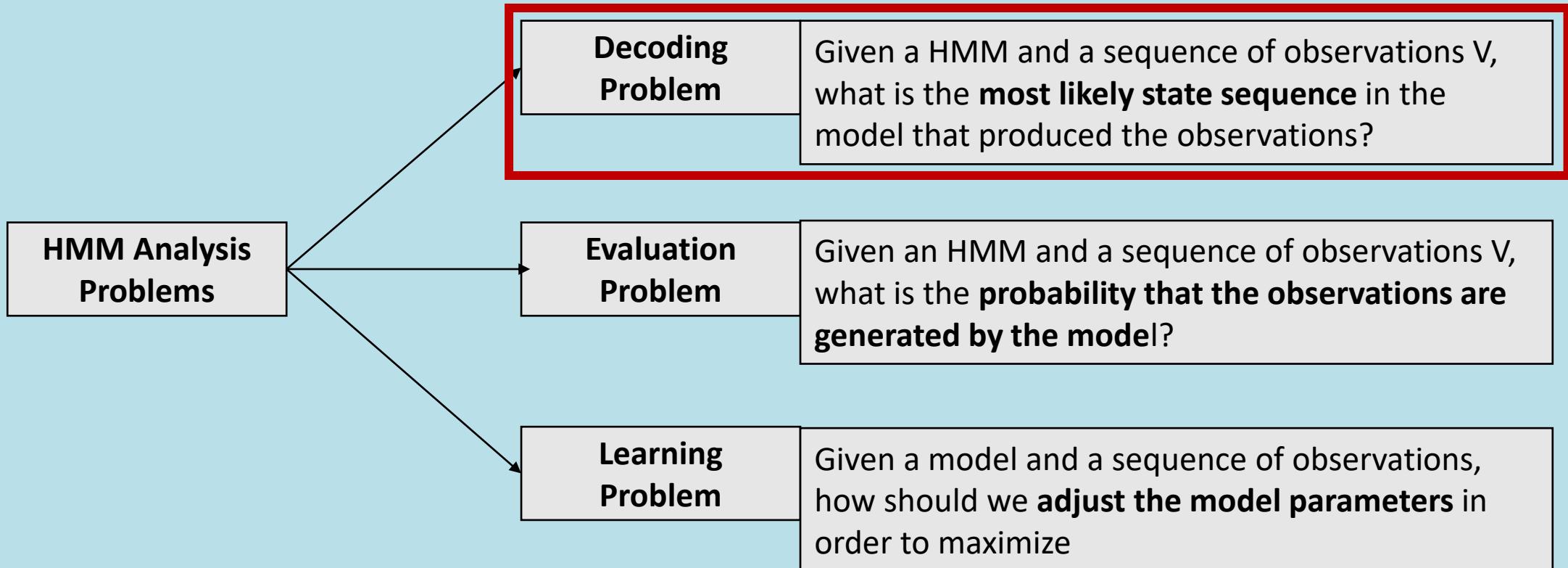
## 8.3 Overview: Values of the Markov Model

	Acronym	Levels
<b>States</b>	$S = \{s_1 \dots s_M\}$ or $I = \{i_1 \dots i_M\}$	Sun, Clouds, Rain
<b>Transition probability matrix</b>	$P = \{p_{ij}\}$	0.5 0.25 0.25 0.375 0.125 0.5 0.125 0.675 0.2
<b>Emission probability matrix</b>	$B = \{b_{ij}\}$	0.6 0.2 0.15 0.05 0.25 0.25 0.25 0.25 0.05 0.1 0.35 0.5
<b>Start probabilities</b>	$\pi_0$	0.63 0.17 0.2
<b>Observations</b>	O	Dry, Rather dry, Rather wet, Wet
<b>Actual observations</b>	Y	Dry, dry, dry, wet, ratherdry, dry, ...

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Common Problems in Sequence-Analysis

- A popular paper by Rabiner (1989) introduced the idea that HMM should be characterized by three fundamental problems:



Adapted from Rabiner LR (1989); Ferguson J (1960)

### ■ Decoding Problem

- Given a HMM and a sequence of observations  $O$ , what is the most likely state sequence in the model that produced the observations?

### ■ Given a sequence of observations

- $O = Dry, Rather\ dry, Rather\ wet, Wet$

- What portion of the sequence was generated from the sunny weather, what portion from the cloudy day, and what portion from the rainy day?



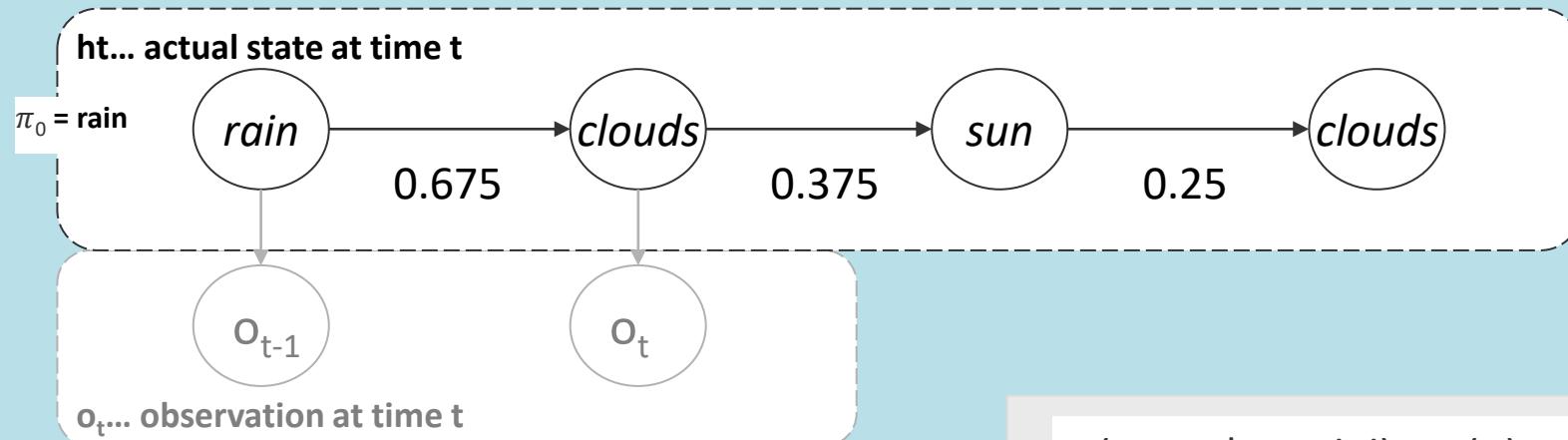
- $O = Dry, rather\ dry, rather\ wet, wet, ...$

## 8.3 HMM: How to read it



How likely is it that on the three days following a rainy day the sequence "clouds, sun, clouds" will occur?

	Sun	Clouds	Rain
Sun	0.5	<b>0.25</b>	0.25
Clouds	<b>0.375</b>	0.125	0.5
Rain	0.125	<b>0.675</b>	0.2



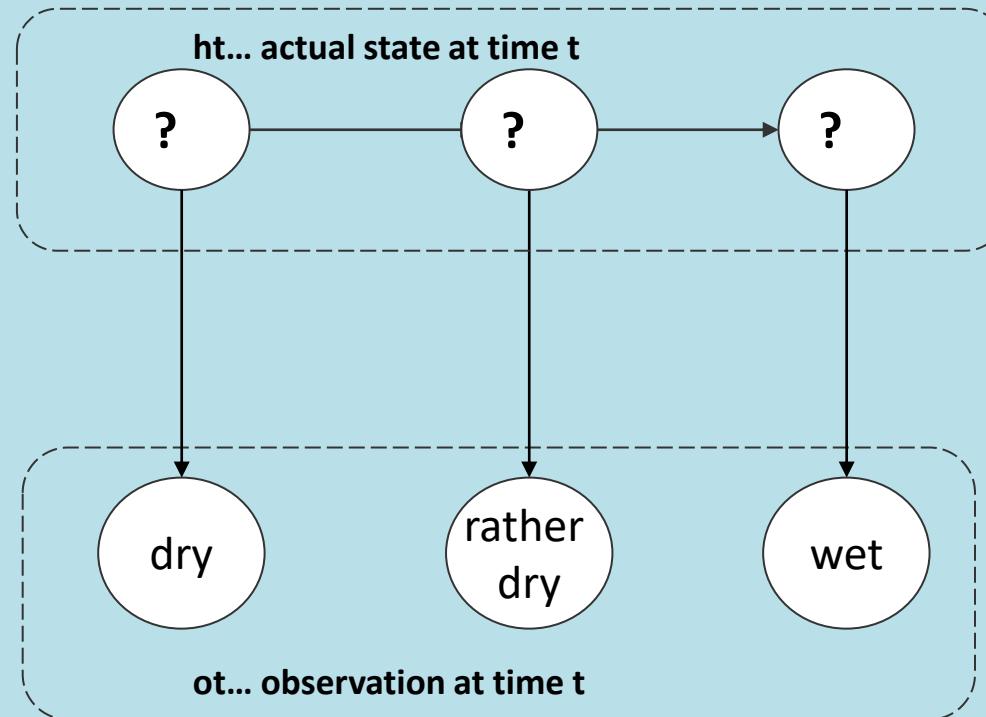
$$\begin{aligned} P(\text{RCSC} \mid \text{Model}) &= P(R) \cdot P(C|R) \cdot P(S|C) \cdot P(C|S) \\ P(\text{RCSC} \mid \text{Model}) &= 1 \cdot 0.675 \cdot 0.375 \cdot 0.25 \\ P(\text{RCSC} \mid \text{Model}) &= 0.06328125 \end{aligned}$$

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Example 2: A Common Decoding Problem



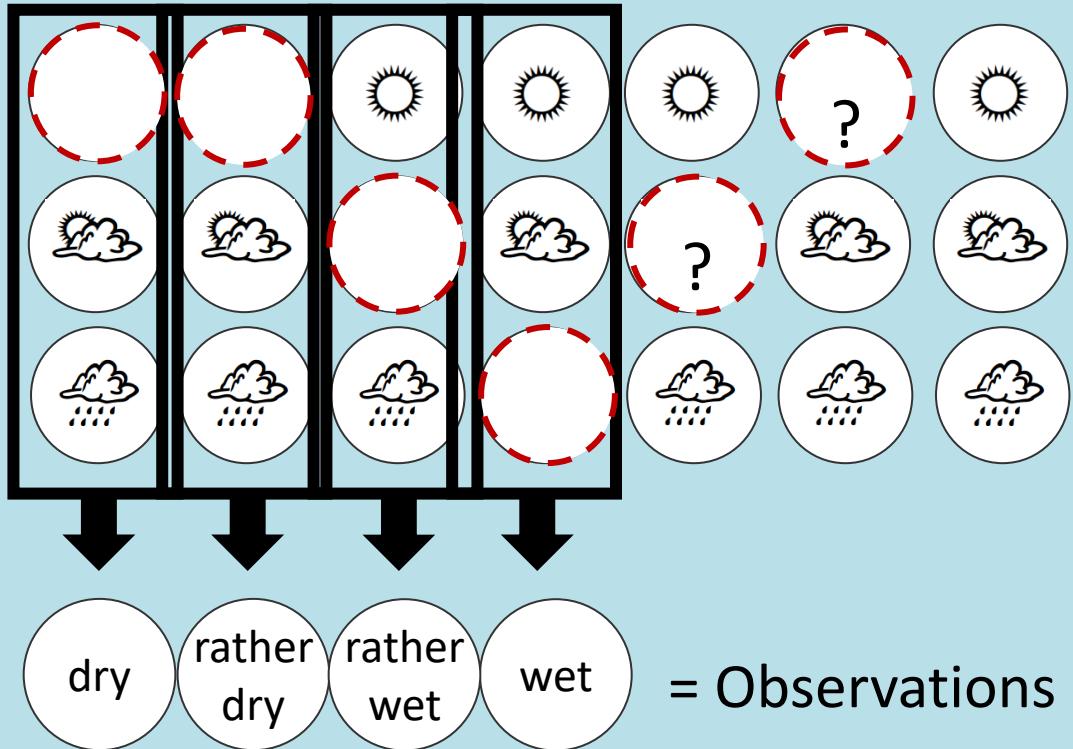
Which is the most probable hidden state sequence if the observed sequence is "dry, rather dry, wet"?



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 The Viterbi Algorithm: Main idea

General principle: Dynamic programming



### Algorithm: Viterbi

```
for each state  $i = 1, 2, \dots, N$ 
     $T_1[i, 1] \leftarrow \pi_i \cdot B_{iy_1}$ 
     $T_2[i, 1] \leftarrow 0$ 
end
for each observation  $j = 2, 3, \dots, T$ 
    foreach state  $i = 1, 2, \dots, N$ 
         $T_1[i, j] \leftarrow \max_k (T_1[k, j - 1] \cdot P \cdot B)$ 
         $T_2[i, j] \leftarrow \operatorname{argmax}_k (T_1[k, j - 1] \cdot P \cdot B)$ 
    end
end
 $z_T \leftarrow \operatorname{argmax}_k (T_1[k, T])$ 
 $x_T \leftarrow s_{zT}$ 
for  $j = T, T - 1, \dots, 2$ 
     $z_{j-1} \leftarrow T_2[z_j, j]$ 
     $x_{j-1} \leftarrow s_{z_{j-1}}$ 
end
```

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.3 Example 2 ► State 1

- Start probabilities  $\pi$

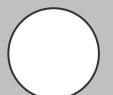
<i>sun</i>	0.63
<i>clouds</i>	0.17
<i>rain</i>	0.20

- Sequence:  
„dry, rather dry,  
wet“

- Emission  
probabilities  $B$ :

	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05
<i>clouds</i>	0.25	0.25	0.25	0.25
<i>rain</i>	0.05	0.1	0.35	0.5

*sun*



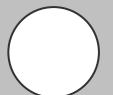
$$0.63 \cdot 0.6 = 0.378$$

*clouds*



$$0.17 \cdot 0.25 = 0.0425$$

*rain*



$$0.2 \cdot 0.05 = 0.01$$

**dry**

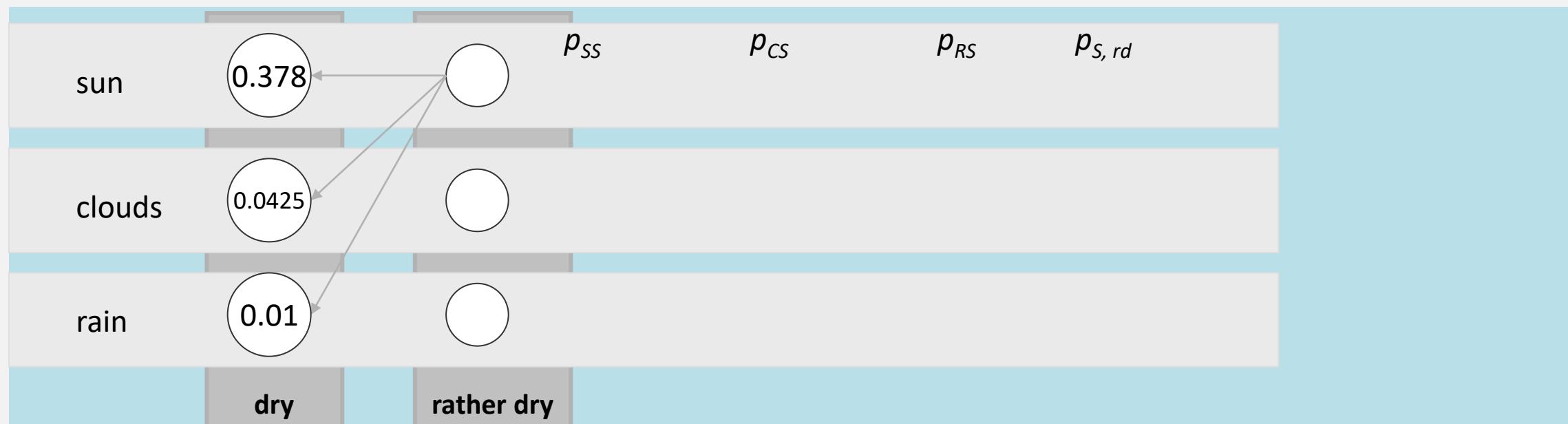
## 8.3 Example 2 ► State 2 = "sun"

- transition probabilities  $P$

<i>after ▶</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>follows ▼</i>			
<i>sun</i>	0.5	0.25	0.25
<i>clouds</i>	0.375	0.125	0.5
<i>rain</i>	0.125	0.675	0.2

- emission probabilities  $B$

<i>If ▶</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>then ▼</i>				
<i>sun</i>	0.6	0.2	0.15	0.05
<i>clouds</i>	0.25	0.25	0.25	0.25
<i>rain</i>	0.05	0.1	0.35	0.5



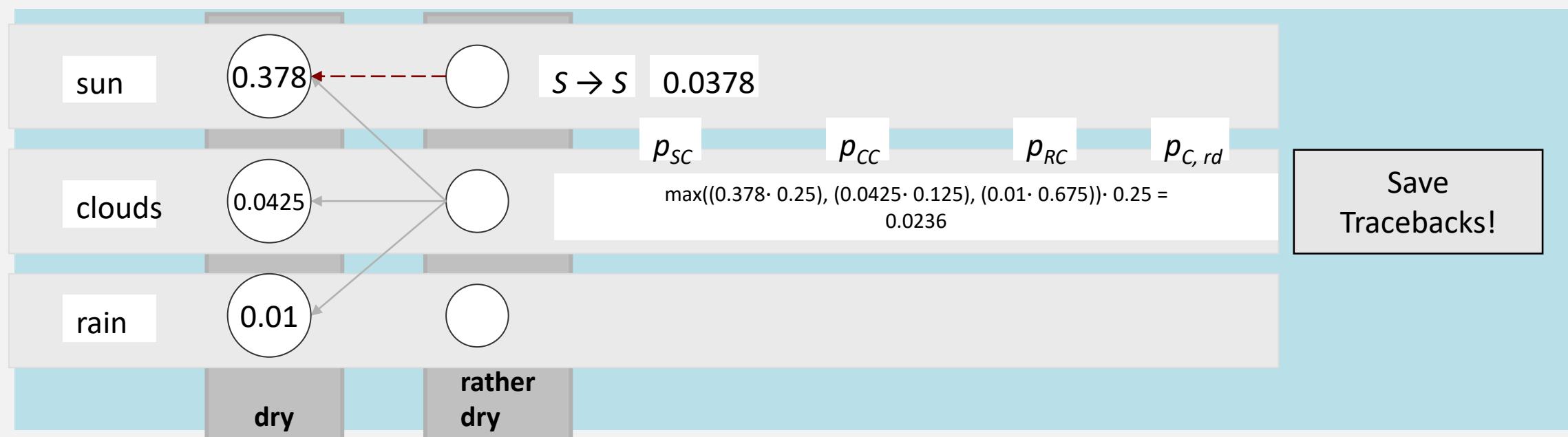
## 8.3 Example 2 ► State 2 = "clouds"

- transition probabilities  $P$

<i>after ▶ follows ▼</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>sun</i>	0.5	0.25	0.25
<i>clouds</i>	0.375	0.125	0.5
<i>rain</i>	0.125	0.675	0.2

- emission probabilities  $B$

<i>If ▶ then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05
<i>clouds</i>	0.25	0.25	0.25	0.25
<i>rain</i>	0.05	0.1	0.35	0.5



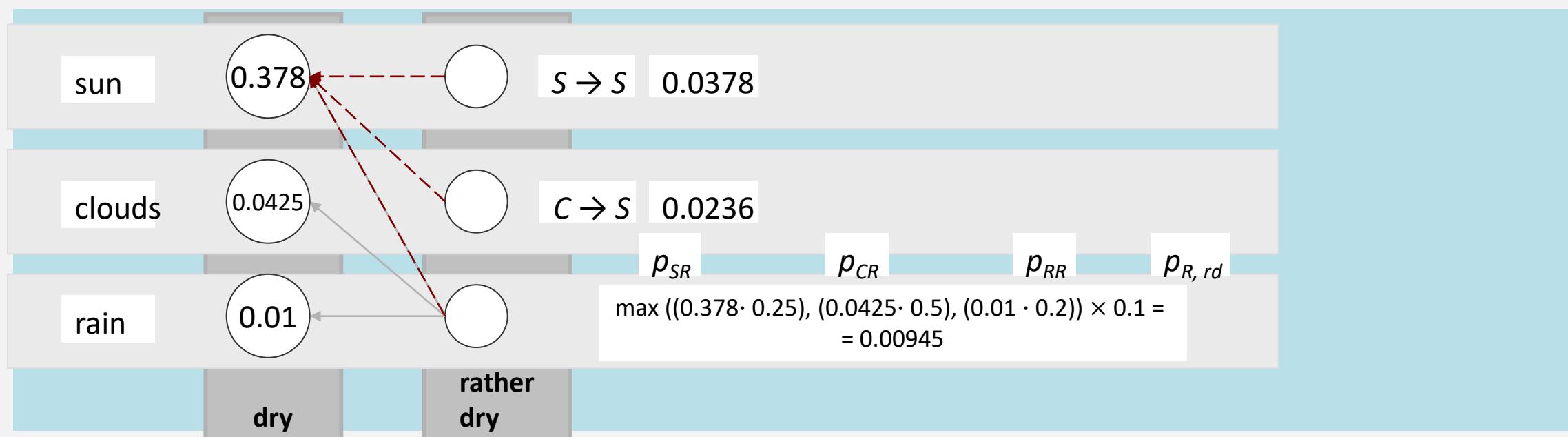
## 8.3 Example 2 ► State 2 = "rain"

- transition probabilities P

	after ►	sun	clouds	rain
	follows ▼			
sun		0.5	0.25	0.25
clouds		0.375	0.125	0.5
rain		0.125	0.675	0.2

- emission probabilities B

	If ►	dry	rather dry	rather wet	wet
	then ▼				
sun		0.6	0.2	0.15	0.05
clouds		0.25	0.25	0.25	0.25
rain		0.05	0.1	0.35	0.5



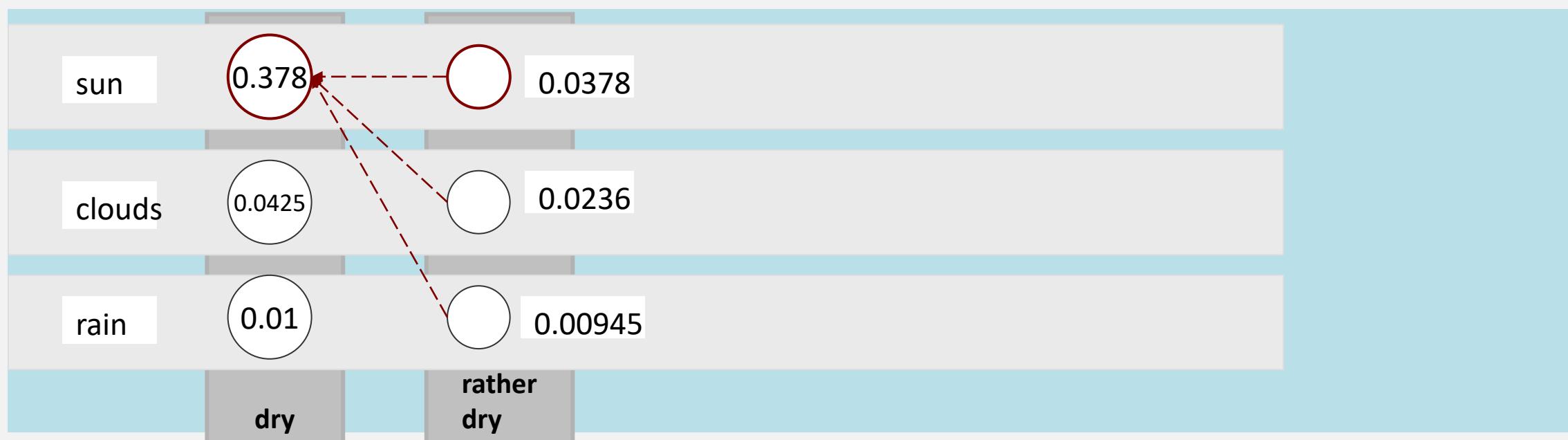
## 8.3 Example 2 ► path(state<sub>1</sub>, state<sub>2</sub>)

- transition probabilities P

	<i>after ▶ follows ▼</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>sun</i>	0.5	0.25	0.25	
<i>clouds</i>	0.375	0.125	0.5	
<i>rain</i>	0.125	0.675	0.2	

- emission probabilities B

	<i>If ▶ then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05	
<i>clouds</i>	0.25	0.25	0.25	0.25	
<i>rain</i>	0.05	0.1	0.35	0.5	



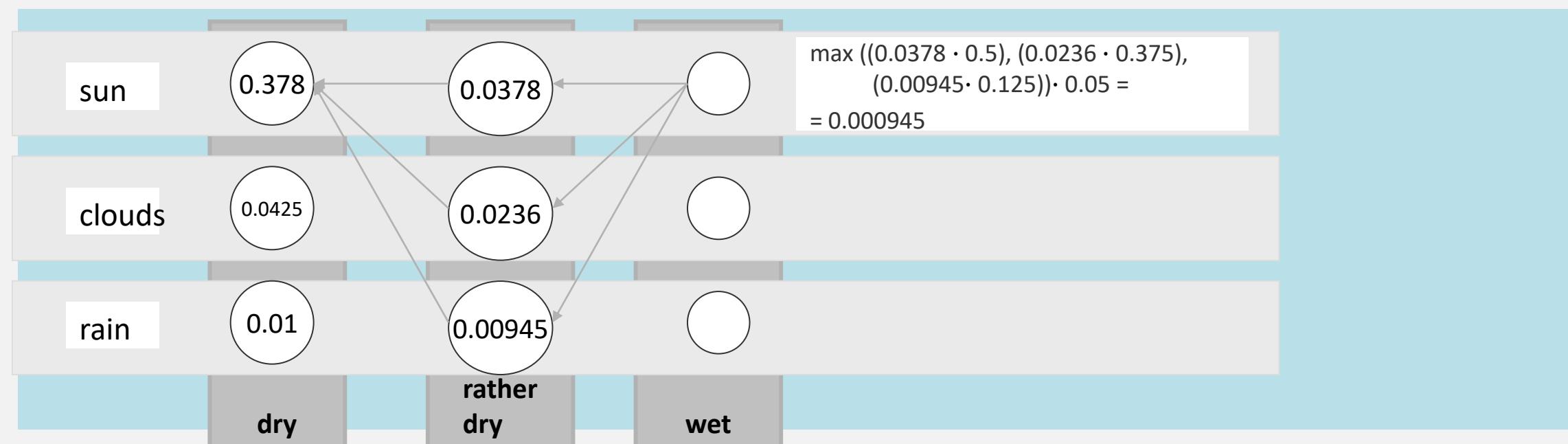
## 8.3 Example 2 ► State 3 = "sun"

- **transition probabilities P**

	<i>after ►</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>follows ▼</i>				
<i>sun</i>	0.5	0.25	0.25	
<i>clouds</i>	0.375	0.125	0.5	
<i>rain</i>	0.125	0.675	0.2	

- **emission probabilities B**

	<i>If ►</i>	<i>then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>						
<i>clouds</i>						
<i>rain</i>			0.05	0.2	0.15	0.05



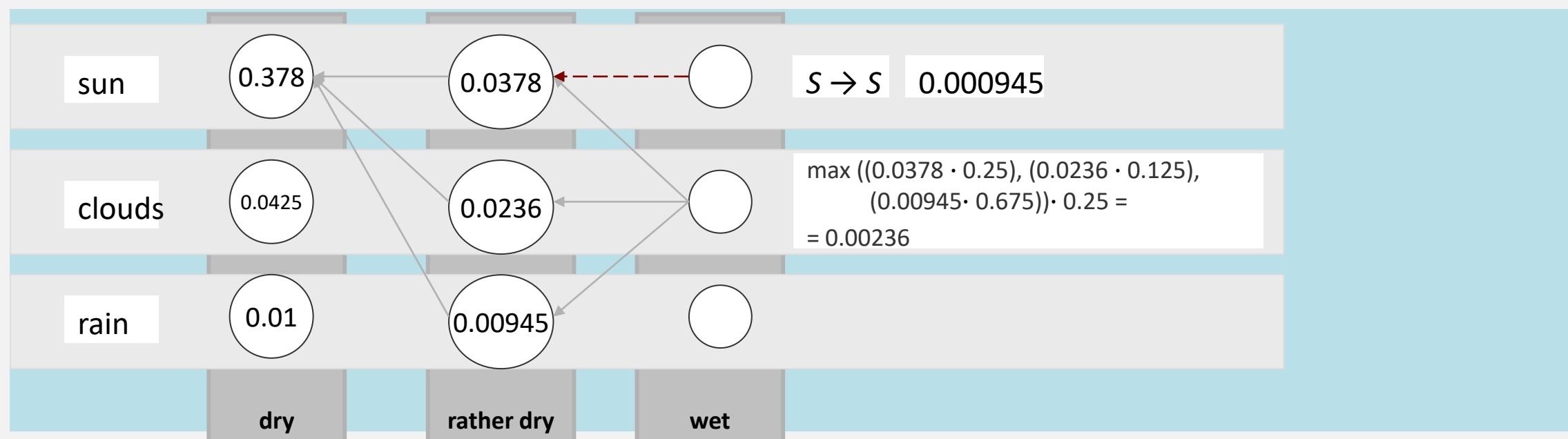
## 8.3 Example 2 ► State 3 = "clouds"

- **transition probabilities P**

<i>after ▶ follows ▼</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>sun</i>	0.5	0.25	0.25
<i>clouds</i>	0.375	0.125	0.5
<i>rain</i>	0.125	0.675	0.2

- **emission probabilities B**

<i>If ▶ then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05
<i>clouds</i>	0.25	0.25	0.25	0.25
<i>rain</i>	0.05	0.1	0.35	0.5



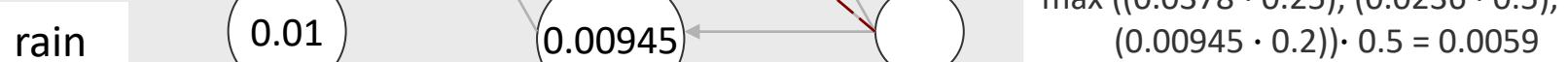
## 8.3 Example 2 ► State 3 = "rain"

▪ transition probabilities P

	<i>after ▶ follows ▼</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>sun</i>	0.5	0.25	0.25	
<i>clouds</i>	0.375	0.125	0.5	
<i>rain</i>	0.125	0.675	0.2	

▪ emission probabilities B

	<i>If ▶ then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05	
<i>clouds</i>	0.25	0.25	0.25	0.25	
<i>rain</i>	0.05	0.1	0.35	0.5	



$$\max ((0.0378 \cdot 0.25), (0.0236 \cdot 0.5), (0.00945 \cdot 0.2)) \cdot 0.5 = 0.0059$$

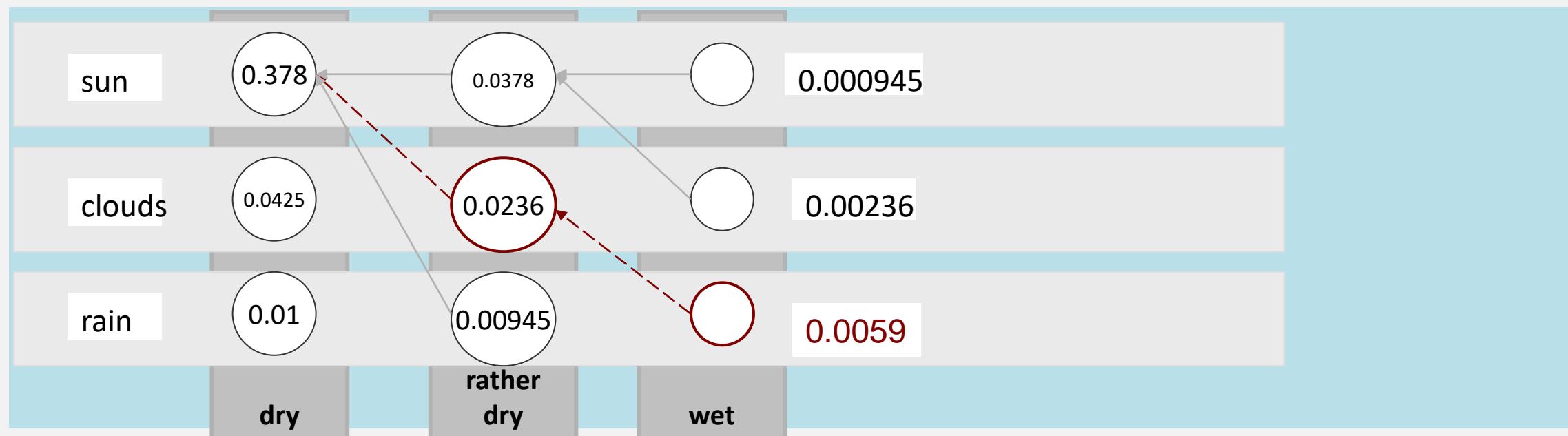
## 8.3 Example 2 ► path(state<sub>2</sub>, state<sub>3</sub>)

▪ transition probabilities P

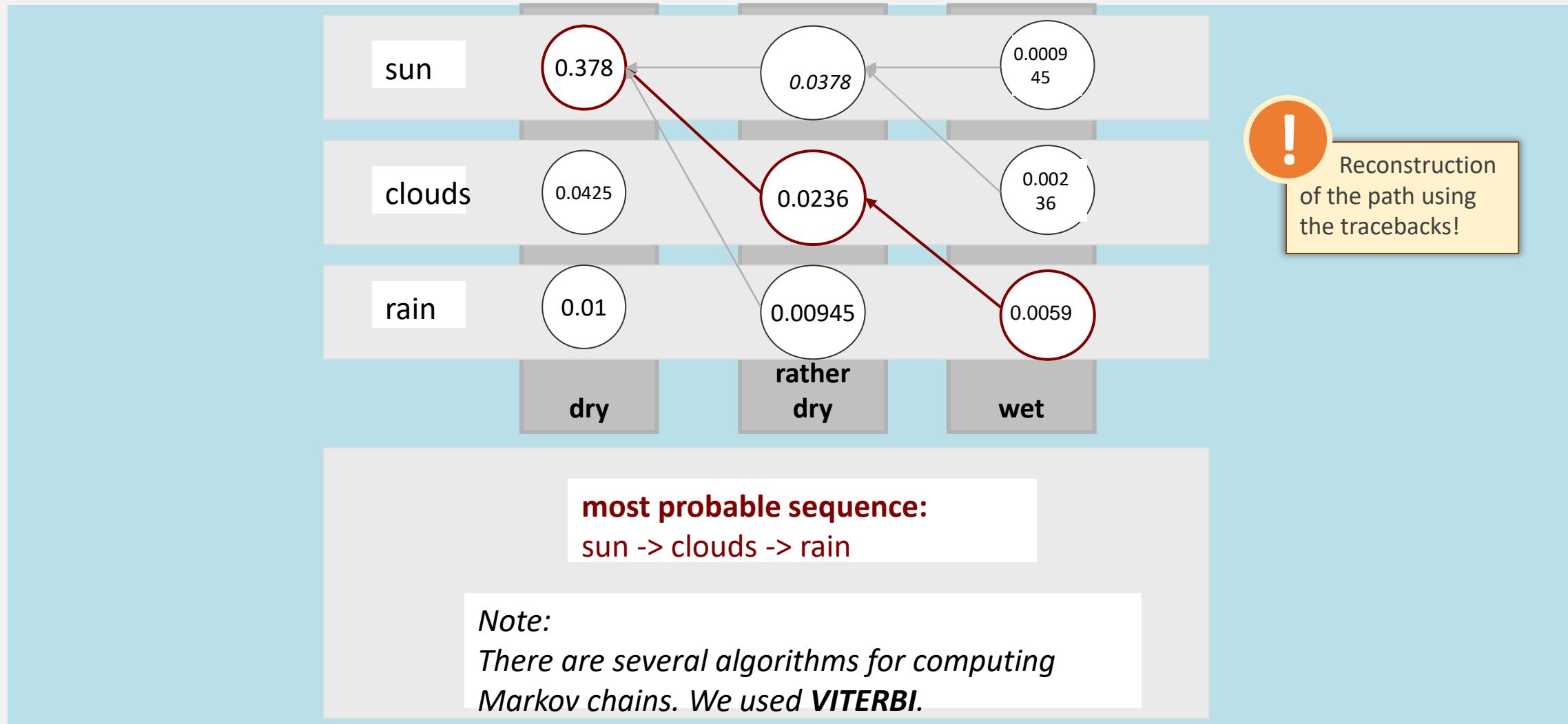
	<i>after ▶ follows ▼</i>	<i>sun</i>	<i>clouds</i>	<i>rain</i>
<i>sun</i>	0.5	0.25	0.25	
<i>clouds</i>	0.375	0.125	0.5	
<i>rain</i>	0.125	0.675	0.2	

▪ emission probabilities B

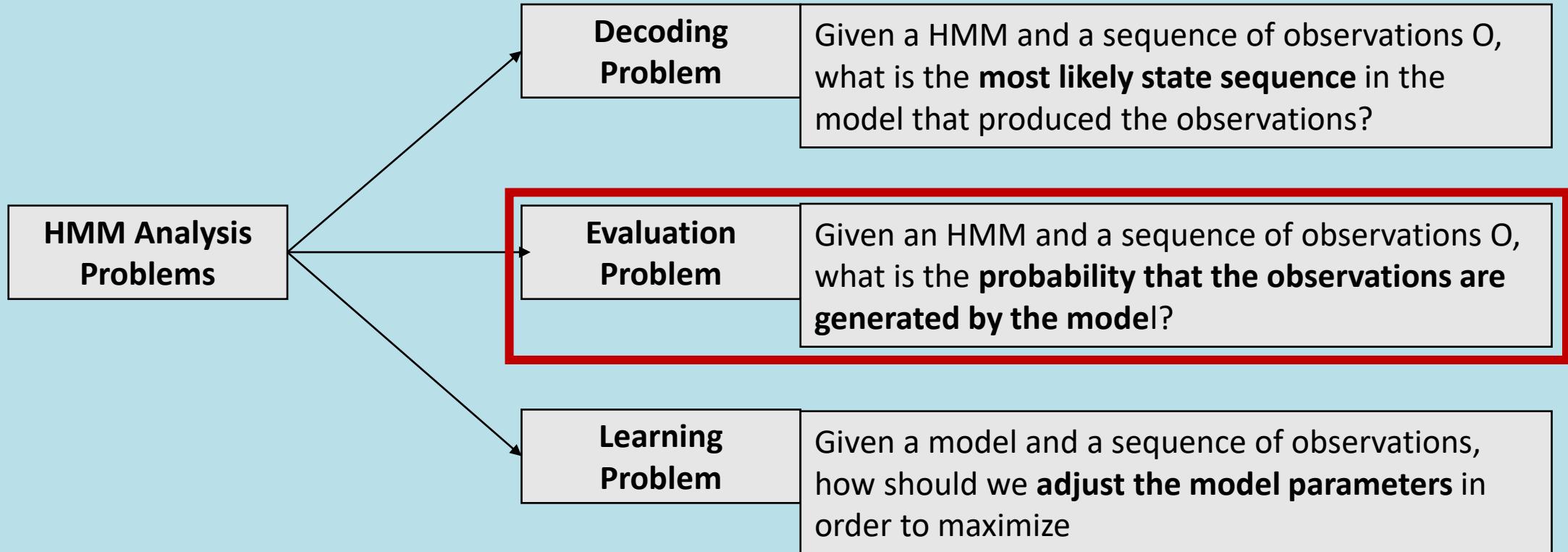
	<i>If ▶ then ▼</i>	<i>dry</i>	<i>rather dry</i>	<i>rather wet</i>	<i>wet</i>
<i>sun</i>	0.6	0.2	0.15	0.05	
<i>clouds</i>	0.25	0.25	0.25	0.25	
<i>rain</i>	0.05	0.1	0.35	0.5	



## 8.3 Example 2 ► path(state<sub>1</sub>, state<sub>2</sub>, state<sub>3</sub>)



## 8.3 Common Problems in Sequence-Analysis



Adapted from Rabiner LR (1989); Ferguson J (1960)

### ■ Evaluation Problem

- Given an HMM and a sequence of observations  $V$ , what is the probability that the observations are generated by the model?

### ■ Given a sequence of observations

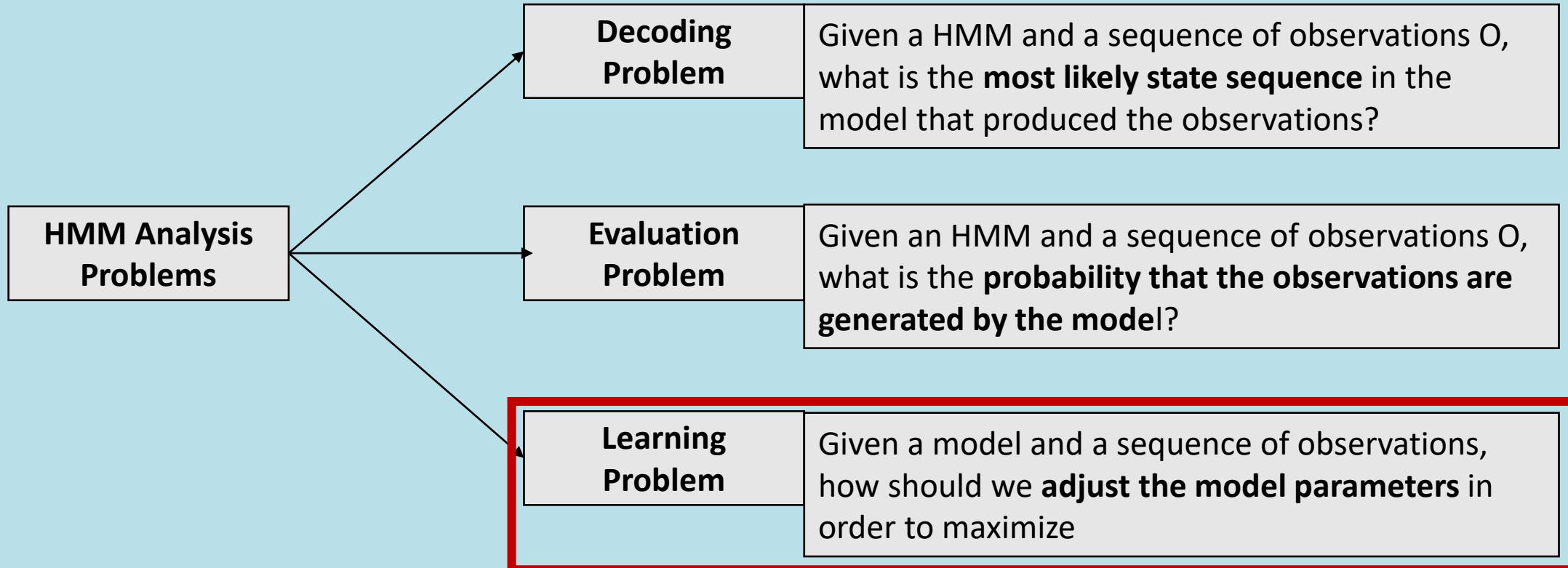
- $V = \{Dry, Rather\ dry, Rather\ wet, Wet\}$

### ■ How likely is this sequence, given our model of how the weather works?



Adapted from Rabiner LR (1989); Ferguson J (1960)

## 8.3 Common Problems in Sequence-Analysis



Adapted from Rabiner LR (1989); Ferguson J (1960)

### ■ Learning Problem

- Given a model and a sequence of observations, how should we adjust the model parameters in order to maximize
- Given a sequence of observations
  - $V = \{Dry, Rather\ dry, Rather\ wet, Wet\}$
- How “rainy” is the rainy day? How “sunny” is the sunny day? How often does the weather change from sunny to rainy, and back?
- Note: We need to know how many states exist

# Your turn!

### Task

What is the basic idea behind the VITERBI algorithm? Which general programming principle does it rely on?

## 8

## Probabilistic Reasoning and Modelling

8.1

From Uncertainty to Probability

8.2

Probabilistic Reasoning

8.3

Probabilistic Reasoning over Time

8.4

Decision Theory and Decision-Making

8.5

Game Theory and Sequential Decision-Making

Lectorial 6: Intelligent Agents in Action

### ► What you will learn:

- Concepts of statistics and probability theory to model agents that can act under uncertainty
- Foundations of Markov theory, Bayes theory and sequence analysis for to reason under uncertainty according to the laws of probability theory
- How to analyze and build optimal agent decision-making that can handle uncertainty

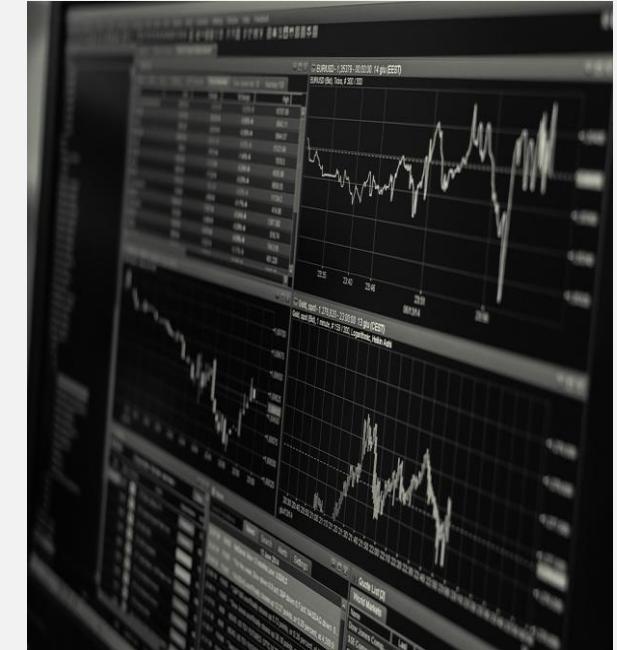


Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

### ► Duration:

- 215 min + 90 min (Lectorial)

### ► Relevant for Exam:

- 8.1 – 8.4

## 8.4 Decision-Making under Uncertainty to Decision Theory

- In chapter 8.1 we defined decision theory as

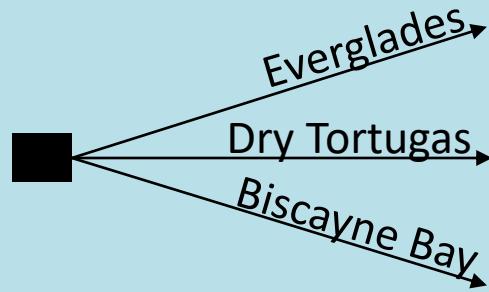
*Decision theory = probability theory + utility theory*



- **Key question:** What is utility, and how can we use it in Artificial Intelligence

## 8.4 Non-Deterministic vs. Probabilistic Uncertainty

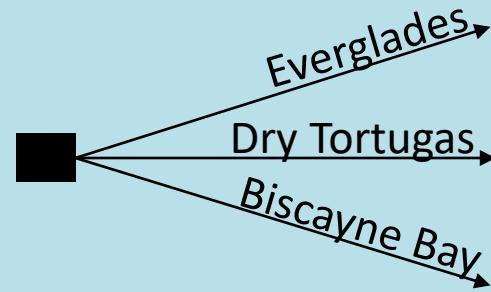
### Non-deterministic model



{ *Everglades,*  
*Dry Tortugas,*  
*Biscayne Bay* }

- Decision that is best for worst case

### Probabilistic model



{  $P(E)$ , *Everglades*;  
 $P(T)$ , *Dry Tortugas*;  
 $P(B)$ , *Biscayne Bay* }

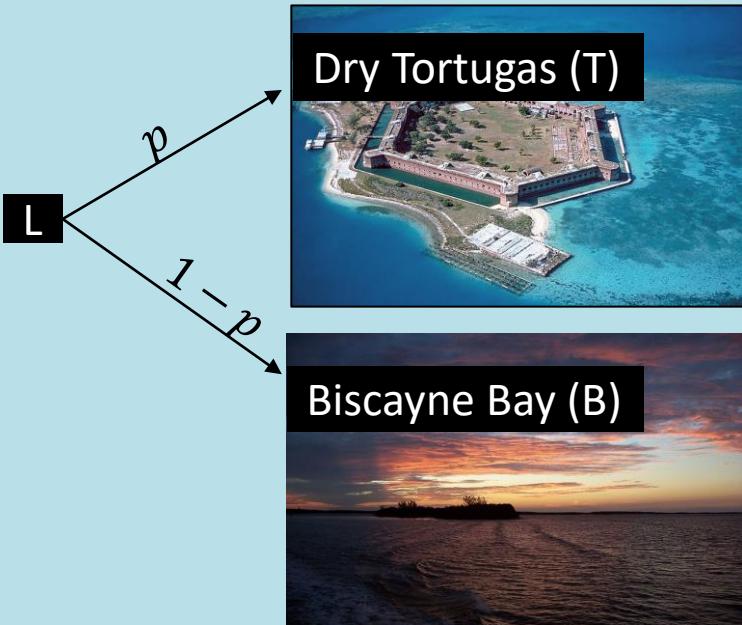
- Decision that maximizes expected utility value (MEU)

- Similar concept than adversarial search

Adapted from Russell, S., & Norvig, P. (2016)

## 8.4 Lotteries and Preferences

- In the following, we call such probabilistic events lotteries
- A lottery consists of a set of events (prizes) and probabilities



$$\text{Lottery } L = [p, T; (1 - p), B]$$

- Preferences
  - An agent likes certain prizes better than others
  - An agent therefore also likes certain lotteries better than others
- Notations:  $T > B$  vs.  $T \sim B$  vs.  $T \geq B$

Adapted from Russell, S., & Norvig, P. (2016) | Image source: ↗[Fort-Jefferson Dry-Tortugas](#)(2005) by U.S. National Park Service from Wikimedia / public domain; ↗[Fort-Jefferson Dry-Tortugas](#)(2005) by U.S. National Park Service from Wikimedia / ↗[CC BY-SA 3.0](#)

## 8.4 Constraints for Preferences in Rational Behavior I

The following six rules define constraints on preferences

- Orderability

$$(A > B) \vee (B > A) \vee (B \sim A)$$

- Transitivity

$$(A > B) \wedge (B > C) \Rightarrow (A > C)$$

- Continuity

$$A > B > C \Rightarrow \exists p [p, A; 1 - p, C] \sim B$$

- Substitutability

$$A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$$

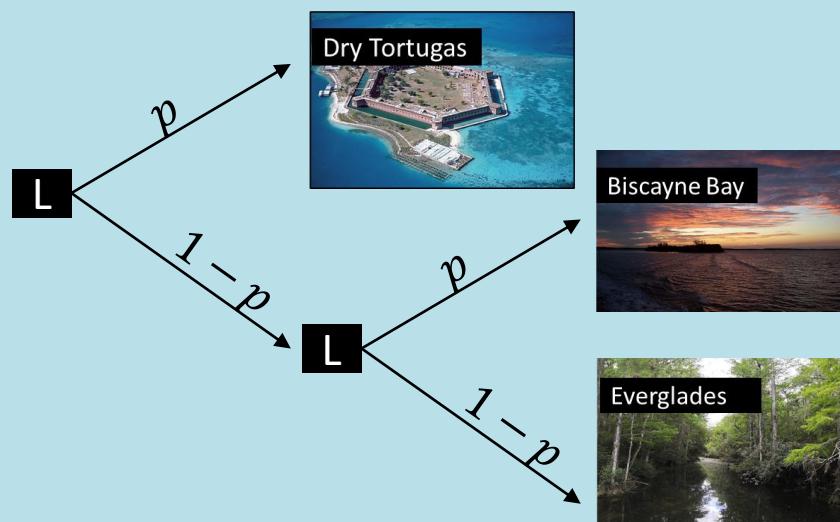
- Monotonicity

$$A > B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1 - p, B] \geq [q, A; 1 - q, B])$$

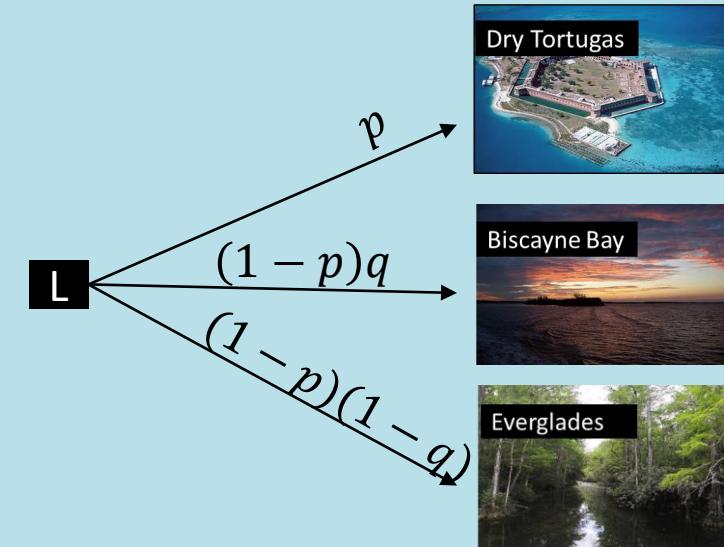
## 8.4 Constraints for Preferences in Rational Behavior II

- Another property that should be obeyed is that lotteries are decomposable
- Therefore, no rational agent should have a preference between the two equivalent formulations

$$[p, T; 1 - p, [q, B; 1 - q, C]] \sim [p, T; (1 - p)q, B; (1 - p)(1 - q), E]$$



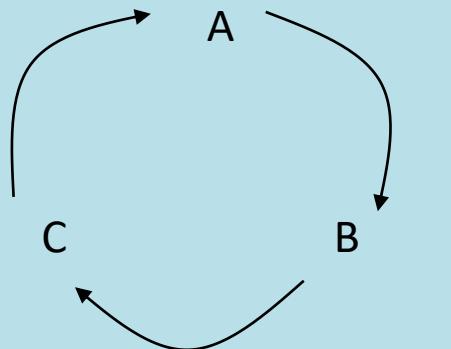
*is equal to*



Adapted from Russell, S., & Norvig, P. (2016) | Image source: ↗[Fort-Jefferson Dry-Tortugas](#)(2005) by U.S. National Park Service from Wikimedia / public domain; ↗[Fort-Jefferson Dry-Tortugas](#)(2005) by U.S. National Park Service from Wikimedia / ↗[CC BY-SA 3.0](#)

## 8.4 Preferences and Irrational Behavior

- Preferences between events (prizes) may be arbitrary
- However, violating the axioms results in irrational behavior
  - E.g. Preferences may be cyclic,  $A > B, B > C, C > A$



- Agents with cyclic preferences will lose all its money in trading games

Adapted from Russell, S., & Norvig, P. (2016)

## 8.4 Utility Functions

- Natural way for measuring how desirable certain prizes are is using utility functions. An utility function assigns a numerical value to each prize
- Utility function naturally lead to preferences

$$A > B \Leftrightarrow U(A) > U(B)$$

- The expected utility of an event is the expected value of the utility function in a lottery

$$EU(X) = \sum_{x \in X} P(x) \cdot U(x)$$

- A utility function in a deterministic environment (no lotteries) is also called a value function

- **Problem:** How to measure Utility
- **Economic workaround:** Measure utility with money
- E.g. If you win 1.000.000€, are you willing to bet them on a double-or-nothing coin flip? What is with a triple or nothing?

$$U(1 \text{ Mio.}) > EU([0.5, 0; 0.5, 3 \text{ Mio.}])$$

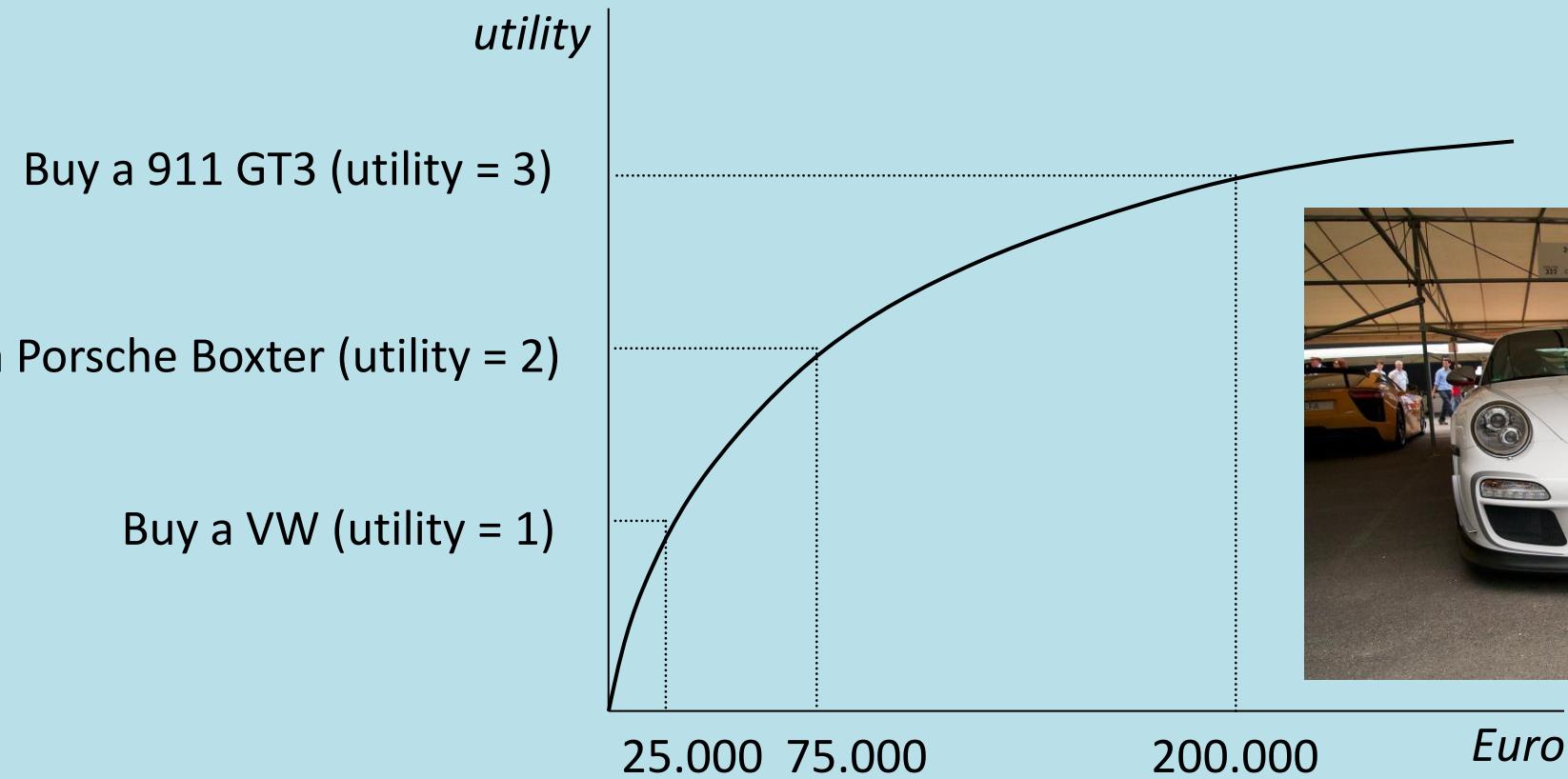
$$U(1 \text{ Mio.}) > 0.5 U(0) + 0.5 U(3 \text{ Mio.})$$



This does not mean that money behaves as a utility function, because it says nothing about preferences between lotteries involving money.

## 8.4 Decreasing Marginal Utility

- Typically, at some point, having an extra dollar does not make people much happier (decreasing marginal utility)

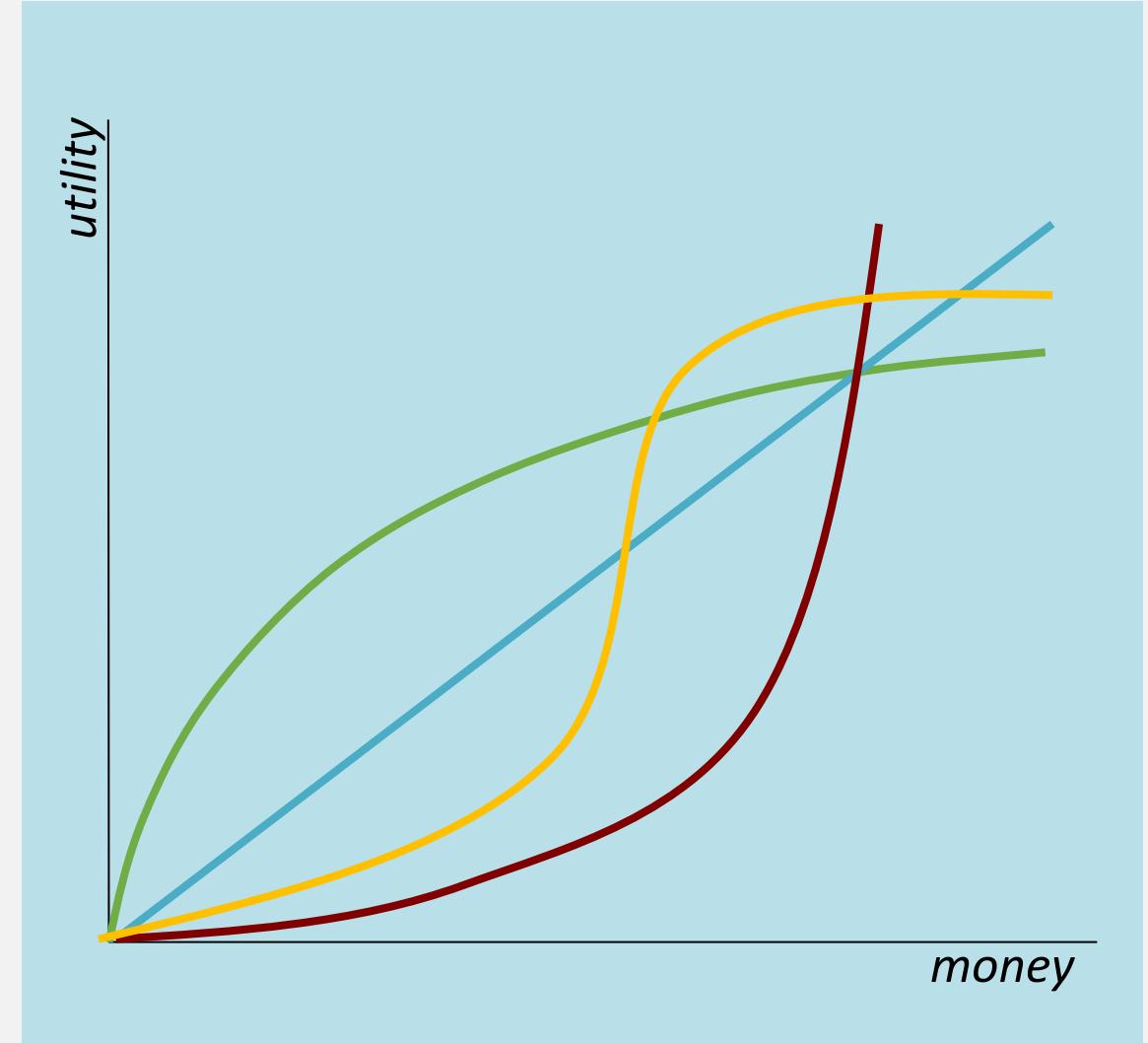


Adapted from Russell, S., & Norvig, P. (2016) Image source: [White Porsche 911 GT3 RS 4.0](#) (2019) by Adam Russell on [Flickr](#) from Wikimedia / [CC BY-SA 2.0](#)

## 8.4 Risk Attitudes under Expected Utility Maximization

### Examples of Risk Attitudes

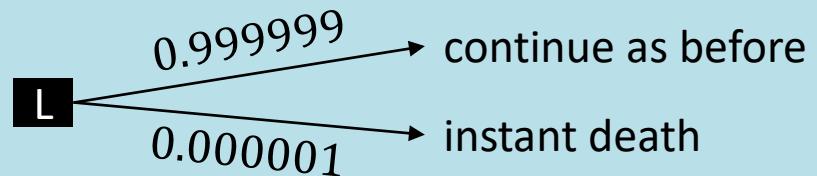
- Green has decreasing marginal utility  
→ risk-averse
- Blue has constant marginal utility  
→ risk-neutral
- Red has increasing marginal utility  
→ risk-seeking
- Yellow's marginal utility is sometimes increasing, sometimes decreasing  
→ neither risk-averse (everywhere) nor risk-seeking (everywhere)



Adapted from Russell, S., & Norvig, P. (2016)

## 8.4 Other Units of Measurements for Utility

- In other domains money is no adequate measurement for utility, e.g. medicine and safety-critical environments
- Quality-Adjusted Life Year (QALY)
- Micromort is the lottery of dying with a probability of one in a million



- Research suggest that a micromort is worth about 50 €

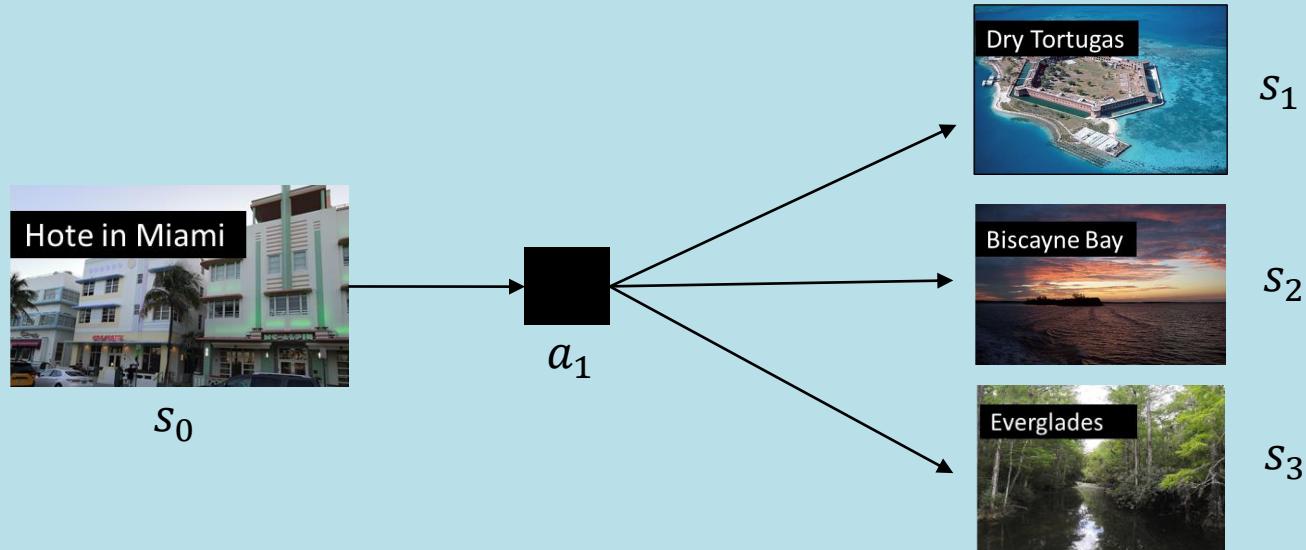
Adapted from Russell, S., & Norvig, P. (2016);

## 8.4 Example 1: Maximizing Expected Roadtrip Utility



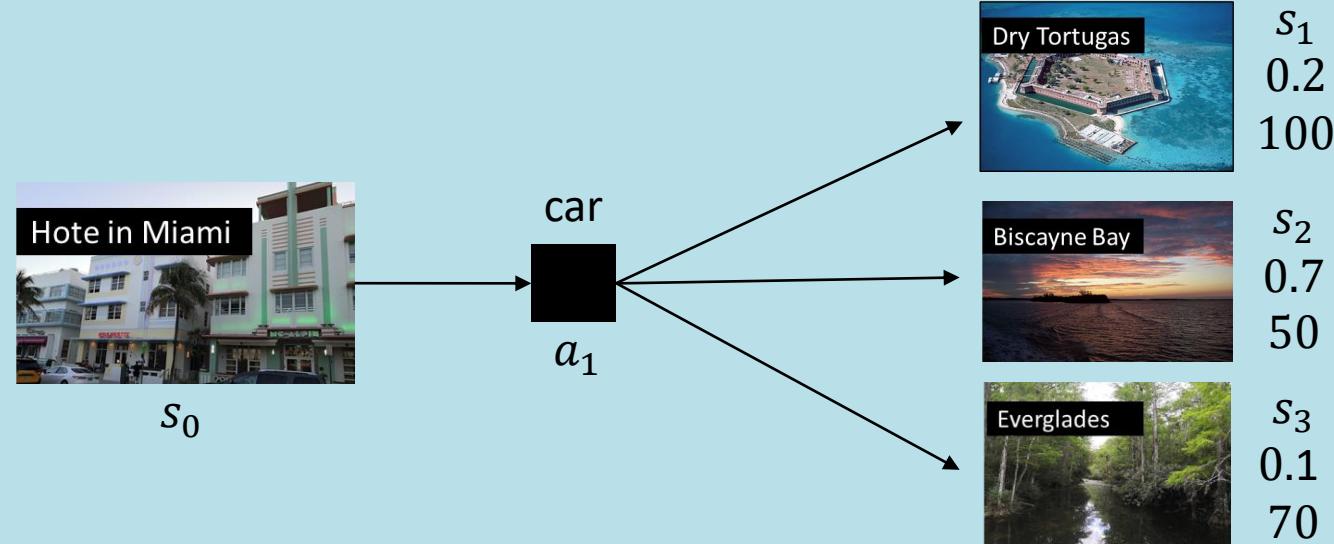
### Maximizing Expected Utility principle (MEU):

A rational agent should choose the action that maximizes agent's expected utility:  $a^* = \arg \max_a EU(a | e)$



Adapted from Russell, S., & Norvig, P. (2016);

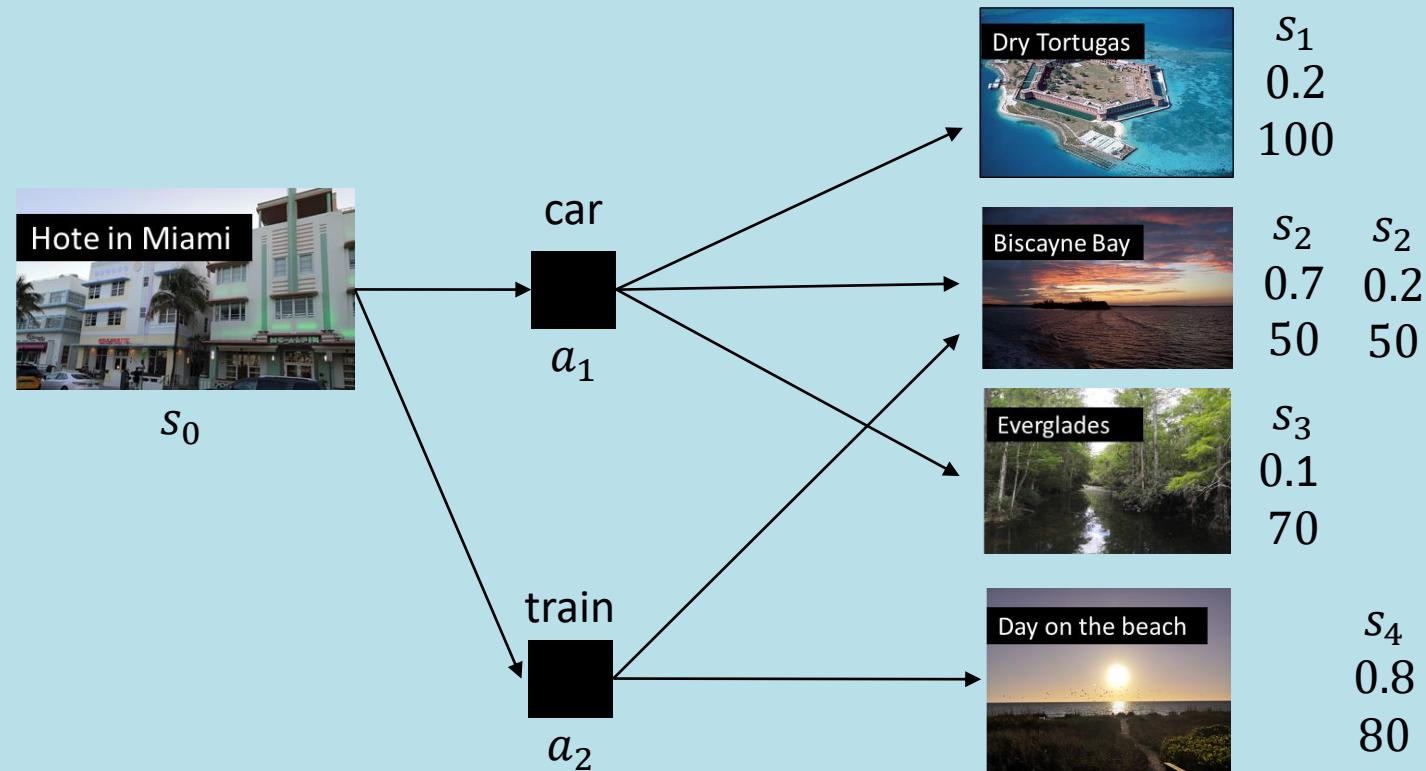
## 8.4 Example 1 ► Expected Utility of an Action



$$\begin{aligned}U(\text{Hotel}) &= 100 \cdot 0.2 + 50 \cdot 0.7 + 70 \cdot 0.1 \\&= 20 + 35 + 7 \\&= 62\end{aligned}$$

Adapted from Russell, S., & Norvig, P. (2016);

## 8.4 Example 1 ► Choice Between 2 Actions

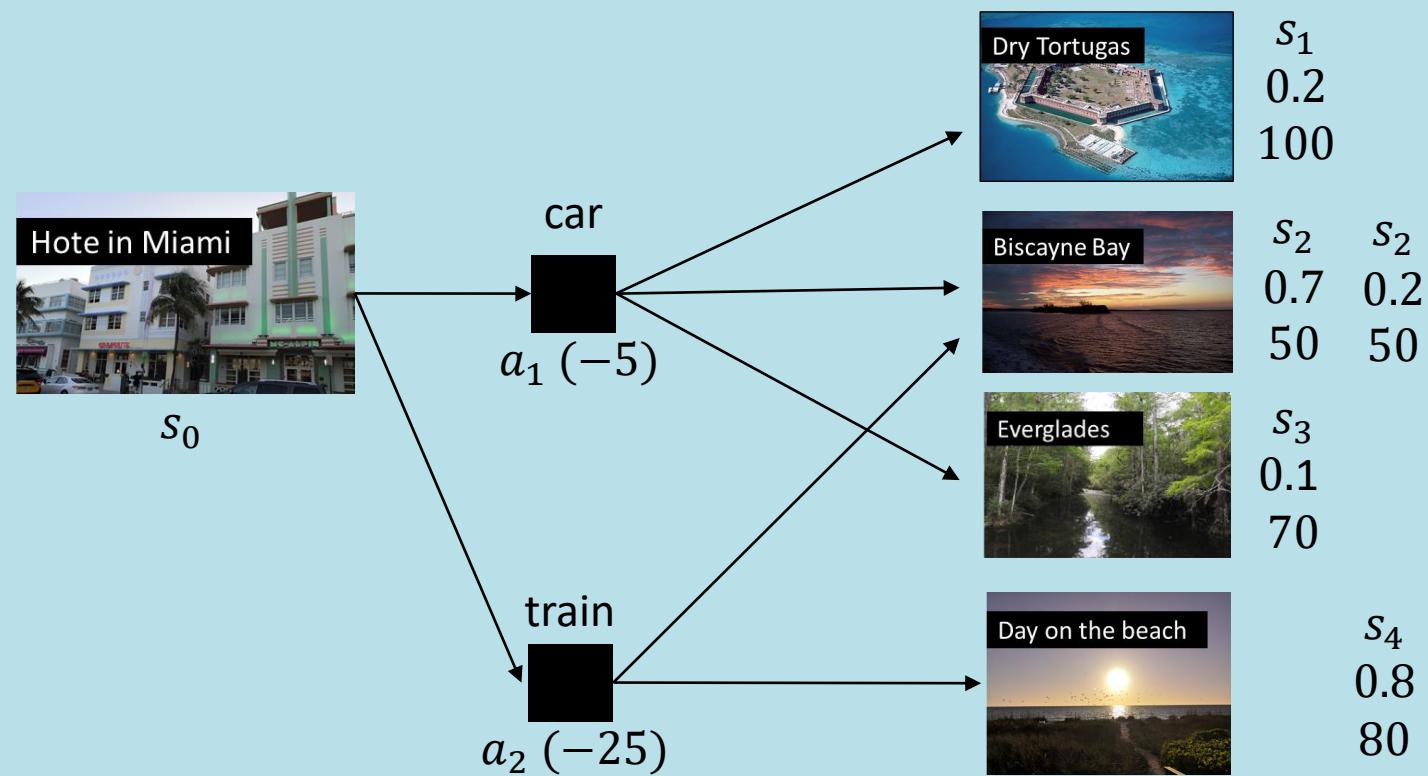


$$U_1(\text{Hotel}) = 62$$
$$U_2(\text{Hotel}) = 74$$

$$U(\text{Hotel}) = \max\{U_1(\text{Hotel}), U_2(\text{Hotel})\}$$
$$U(\text{Hotel}) = 74 \rightarrow \text{go by train } (a_2)$$

Adapted from Russell, S., & Norvig, P. (2016);

## 8.4 Example 1 ► Adding Action Costs



$$U_1(\text{Hotel}) = 62 - 5 = 57$$

$U(\text{Hotel}) = 57 \rightarrow \text{pick driving by car } (a_1)$

$$U_2(\text{Hotel}) = 74 - 25 = 49$$

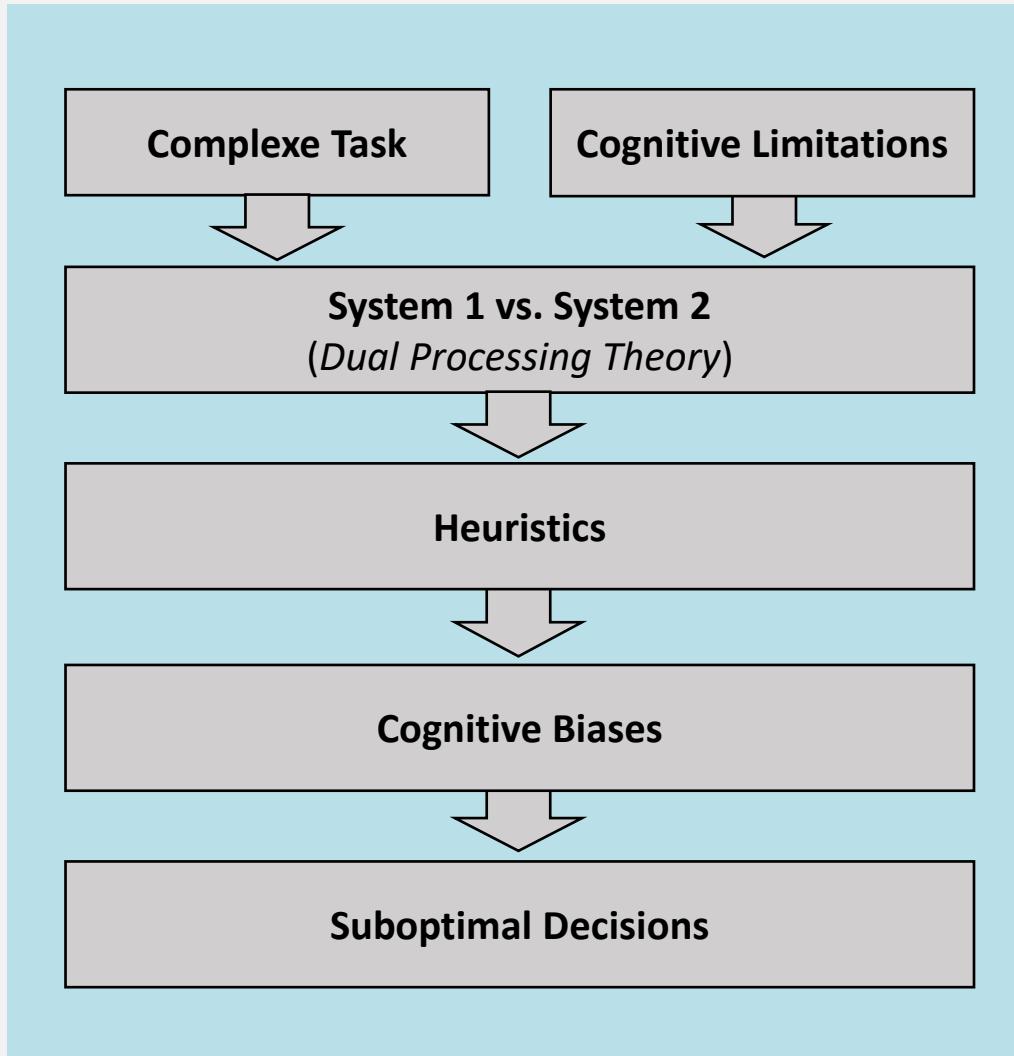
Adapted from Russell, S., & Norvig, P. (2016);

## 8.4 MEU and Artificial Intelligence

- A rational agent should choose the action that maximizes the expected utility
  - The MEU principle provides a normative criterion for rational choice of action
  - Do we now have a working definition of rational behavior, and therefore solved/generated true artificial intelligence?
- 
- Not quite, we have to
    - Build a complete model (actions, utilities, states)
    - Even if you have a complete model, it will be computationally intractable
  - In fact, a truly rational agent takes into account the utility of reasoning as well
    - bounded rationality

Adapted from Russell, S., & Norvig, P. (2016);

## 8.4 Bounded Rationality



Adapted from Jung D. (2019); Kahneman D. (2012); Gigerenzer G. & Selten R.(2002)

“I distinguish between  
**two cognitive systems**, a  
fast system 1 and a slow  
system 2”



Daniel Kahneman,

“maximizing with the goal  
of satisficing, of finding a  
course of action that is  
**good enough**”



Herbert Simon

“In general, these heuristics  
are quite useful, but  
sometimes they **lead to**  
**severe and systematic**  
**errors.**”



Amos Tversky,  
Daniel Kahneman,

# Business Case: Robo Advisors in Financial Consulting (2016)

**CNN Money** International + Markets Economy Companies Tech Autos India

## 30% of bank jobs are under threat

by Matt Egan @MattEganCNN

April 4, 2016: 11:20 AM ET



Message from Davos: The robots are coming

**Robo-Advisor**  
Der Roboter verliert seine Unschuld

### Robo-advisors: The future of investing or the latest financial craze?

## Robo-Advisor - vertrauen Sie einem Computer?

Internet und Geldanlage (Teil 2)

### Robo-Advisor: Reich mit System?

KOLUMNE | Verkehrte (Finanz)welt

### Keine Angst vorm Robo-Advisor

Online-Vermögensverwaltung

### Robo-Advisors sammeln Millionen

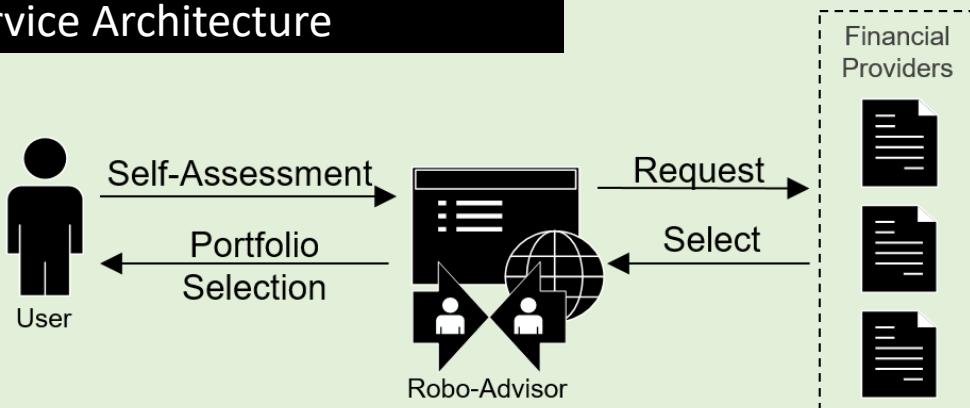
Robo-advisers are trying to tackle the gender-investing gap

# Business Case: Robo Advisors in Financial Consulting

## Robo-Advisors

- Guide users through a self-assessment process and give financial investment advisory
- Robo-advisor computes different investment recommendations and supports individual portfolio selection based on State-of-the-Art artificial intelligence

## Service Architecture



Adapted from Jung D. (2019); Jung D. et al (2018)



- Jung, D., Dorner, V., Glaser, F., & Morana, S. (2018). Robo-advisory. *Business & Information Systems Engineering*, 60(1), 81-86, online available at Springer Link:  
↗<https://link.springer.com/article/10.1007/s12599-018-0521-9>

# Business Case: Robo Advisors in Financial Consulting

## 01 | Executive Summary

*Information systems researchers design and build financial decision support systems based on statistics, artificial intelligence and quantitative finance for private investors. Robo-advisors are such decision support systems aiming to provide independent advice, and support private households in investment decisions and wealth management. Their design and use and thus their ability to support financial decision-making is a challenge for researchers and practitioners.*

## 02 | Solution

- Build information systems that use artificial intelligence and findings from finance to support individual portfolio selection
- In particular, in situations where private investors are likely to make wrong investment decisions.

## Take-Aways

- Intelligent systems can support financial decision-making
- They are a chance for individuals to get high qualitative, low-budget advice

Adapted from Jung D. (2019); Jung D. et al (2018)



## 03 | References

- <https://link.springer.com/article/10.1007/s12599-018-0521-9>

## 8.4 Decision Theory vs. Reinforcement Learning

- Simple decision-making techniques are good for selecting the best action in simple scenarios
- → Reinforcement Learning is concerned with selecting the optimal action in Sequential Decision Problems
- Problems where a sequence of actions has to be taken until a goal is reached.

# Your turn!

### Task

Chris considers four used cars before buying the one with maximum expected utility. Pat considers ten cars and does the same.

- All other things being equal, which one is more likely to have the better car?
- Which is more likely to be disappointed with their car's quality?

## 8

## Probabilistic Reasoning and Modelling

8.1

From Uncertainty to Probability

8.2

Probabilistic Reasoning

8.3

Probabilistic Reasoning over Time

8.4

Decision Theory and Decision-Making

8.5

Game Theory and Sequential Decision-Making

Lectorial 6: Intelligent Agents in Action

► **What you will learn:**

- Concepts of statistics and probability theory to model agents that can act under uncertainty
- Foundations of Markov theory, Bayes theory and sequence analysis for to reason under uncertainty according to the laws of probability theory
- How to analyze and build optimal agent decision-making that can handle uncertainty

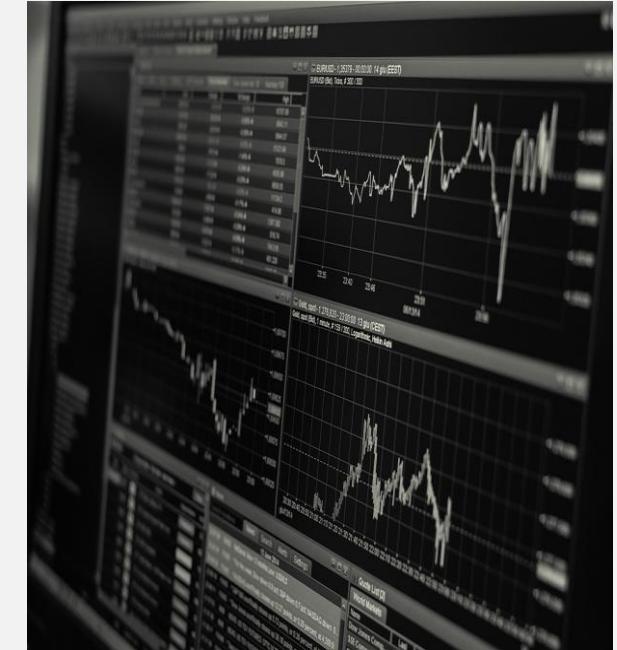


Image source: ↗ [Pixabay](#) (2019) / ↗ [CC0](#)

► **Duration:**

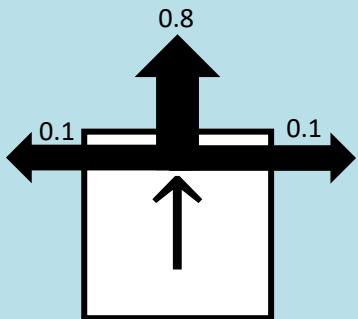
- 215 min + 90 min (Lectorial)

► **Relevant for Exam:**

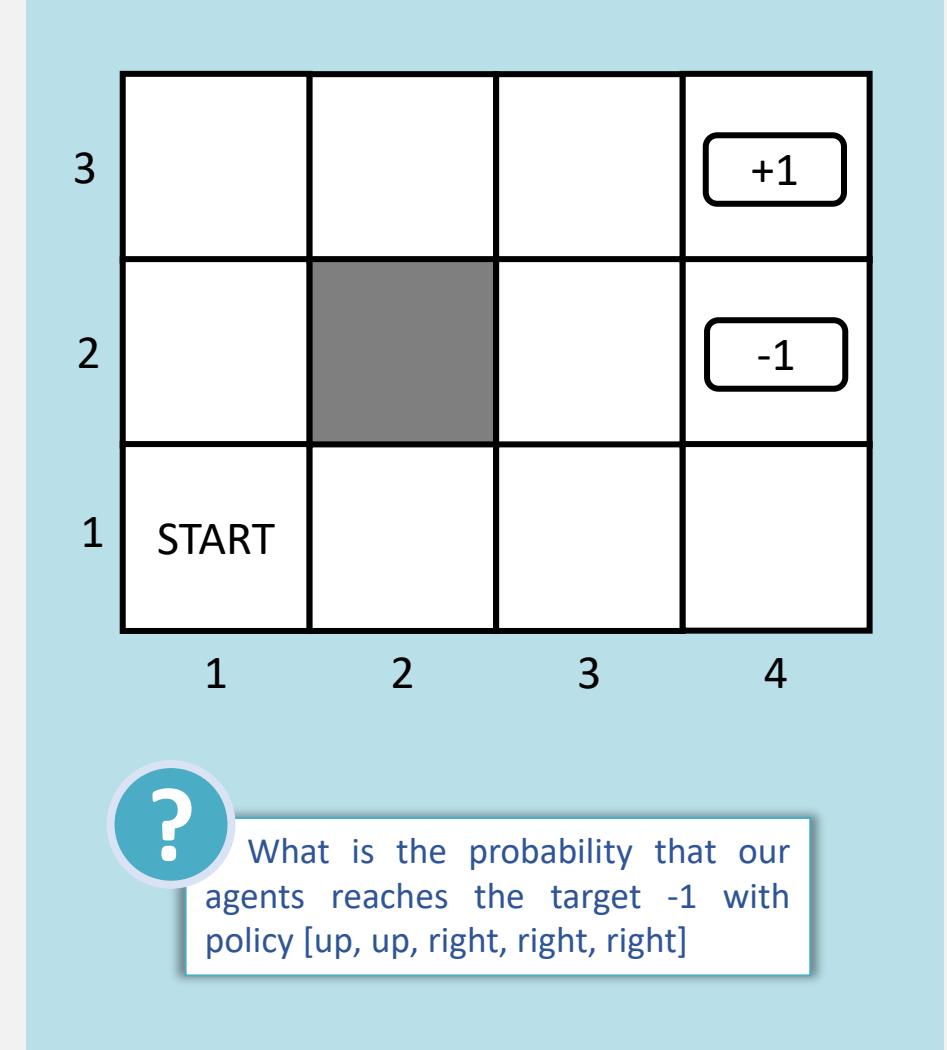
- 8.1 – 8.4

## 8.5 Example 1: Sequential Decision-Making

- Given the following observable environment, where our agent wants to reach  $+1$  or  $-1$
- $\text{Actions}(s) = \{\text{up}, \text{down}, \text{left}, \text{right}\}$
- We have stochastic motion, with the following transition model

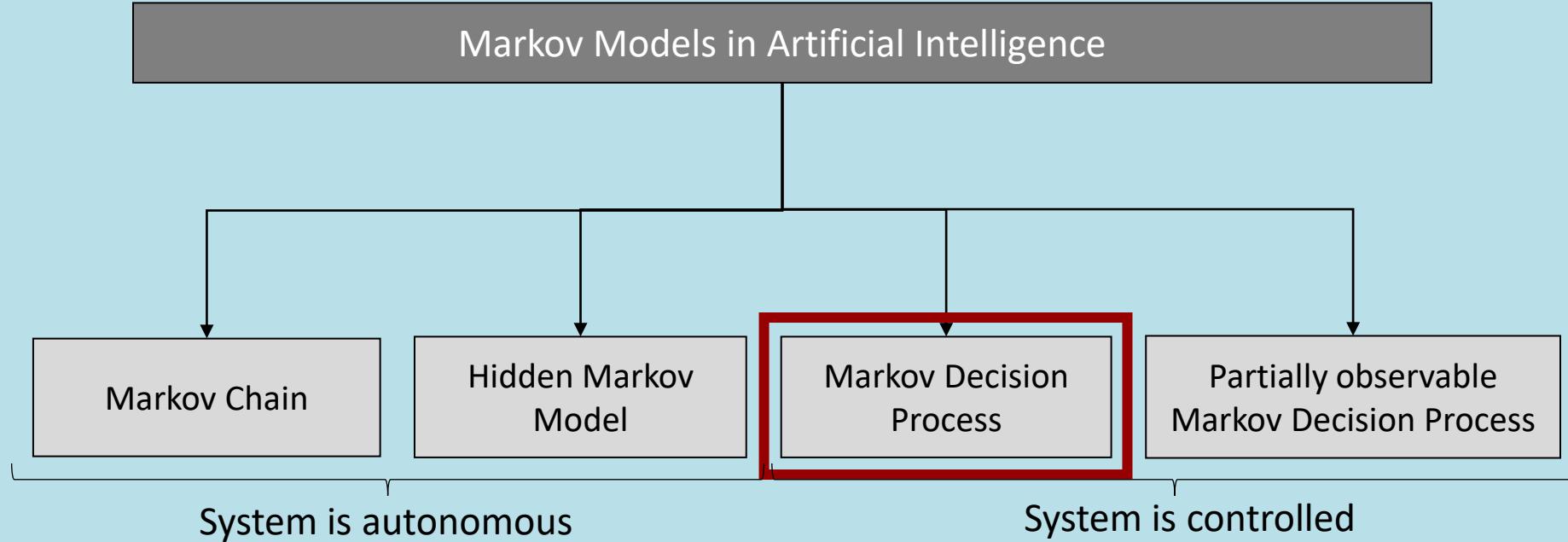


- And rewards  $R(s) = -0.04$  for every non-terminal state



Adapted from Russell, S., & Norvig, P. (2016)

## 8.5 Types of Markov Models – Controlled Markov Models



- Markov process + partial observability = HMM
- Markov process + actions = MDP

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.5 Markov Decision Process (MDP)

- Components:

- States  $s$ , beginning with initial state  $s_0$
- Actions  $a$ , where each state  $s$  has actions  $A(s)$  available from it

- Transition probabilities depend on actions taken

$$P(S_{t+1} = s' | S_t = s, A_t = a) = P(s, a, s')$$

- Rewards for every state, action pair

$$R(S_t = s, A_t = a) = R(s, a)$$



Sometimes people just use  $R(s)$  or  $R(s, a)$ , which is a little more convenient. Discount factor  $\delta$  is sometimes replaced with  $\gamma$ .

- Policy  $\pi(s)$ : the action that an agent takes in any given state

- The “solution” to an MDP (we come later to this)

## 8.5 Maximizing Expected Utility

- In our example defined the utility of a state sequence as the sum of the rewards of the individual states
- The optimal policy should maximize the expected utility over all possible state sequences produced by following that policy
- Expected utility obtained by policy  $\pi$  starting in state  $s$ :

$$U_\pi([s_0, s_1, s_2, \dots]) = R(s_0) + R(s_1) + R(s_2) + \dots$$

- **Problem:** infinite state sequences

## 8.5 Utilities of State Sequences

- Discount the individual state rewards by a factor  $\delta$  between 0 and 1:

$$\begin{aligned} U([s_0, s_1, s_2, \dots]) &= R(s_0) + \delta R(s_1) + \delta^2 R(s_2) + \dots \\ &= \sum_{t=0}^{\infty} \delta^t R(s_t) \leq \frac{R_{\max}}{1-\delta} \end{aligned}$$

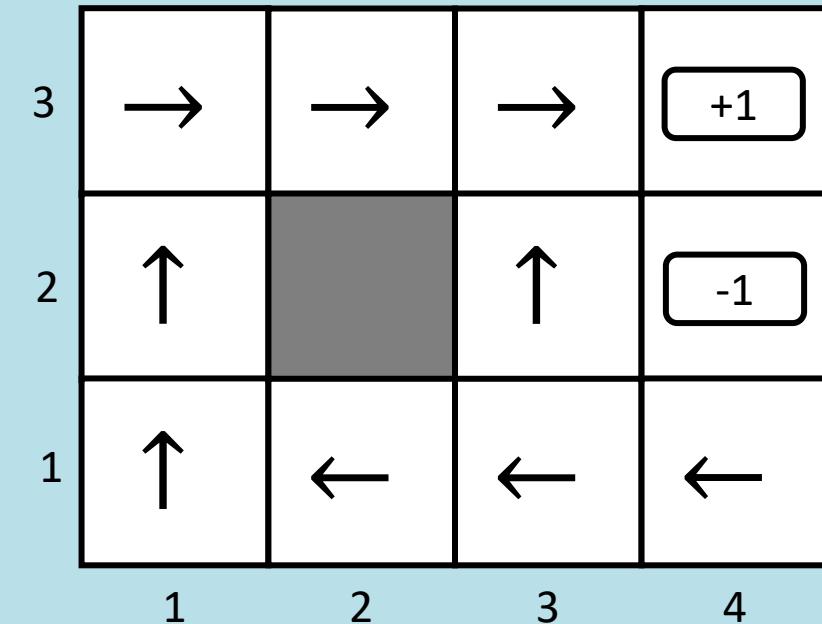
- Sooner rewards count more than later rewards
- Makes sure the total utility stays bounded
- Helps algorithms converge

## 8.5 Optimal Policies

$$U_{\pi}(s) = E \left[ \sum_{t=0}^{\infty} \delta^t R(s_t) \right]$$

$$\pi_s^* = \operatorname{argmax}_{\pi} U^{\pi}(s)$$

$$\pi_s^* = \operatorname{argmax}_{\pi} \sum P(s' | s, a) \cdot U(s')$$



- Optimal policy when  $R(s) = -0.04$  for every non-terminal state

## 8.5 Bellman Equation

- Suppose you are in state  $s$ , and you play optimally from there on
- This leads to expected value  $v^*(s)$

**D**

**Bellman equation:**

$$v^*(s) = \max_a R(s, a) + \delta \sum_{s'} P(s, a, s') v^*(s')$$

- For  $n$  states, we get  $n$  equations in  $n$  unknowns
  - Solving them solves the MDP
  - We could try to solve them through expectimax search, but that would run into trouble with infinite sequences
  - Instead, we solve them algebraically, two methods: **value iteration** and **policy iteration**

## 8.5 Example 1: Value-Iteration

3	0.812	0.868	0.918	+1
2	0.762		0.660	-1
1	0.705	0.655	0.611	0.388

Adapted from Russell, S., & Norvig, P. (2016)

- Find optimal action based on state

$$U(1,1) = -0.04 + \delta \max \{$$

$$0.8 U(1,2) + 0.1 U(2,1) + 0.1 U(1,1) \text{ top}$$

$$0.9 U(1,1) + 0.1 U(1,2) \text{ left}$$

$$0.9 U(1,1) + 0.1 U(2,1) \text{ down}$$

$$0.8 U(2,1) + 0.1 U(1,2) + 0.1 U(1,1) \} \text{ right}$$

## 8.5 Value iteration Algorithm

- Iteratively update values for states using Bellman equation
- $v_i(s)$  is our estimate of value of state  $s$  after  $i$  updates

$$v_{i+1}(s) = R(s) + \delta \max_a \sum_{s'} P(s'| a, s ) v_i(s')$$

- Will converge
- If we initialize  $v_0 = 0$  everywhere, then  $v_i(s)$  is optimal expected utility with only  $i$  steps left (finite horizon)
- Again, dynamic program from the future to the present

## 8.5 Policy Iteration

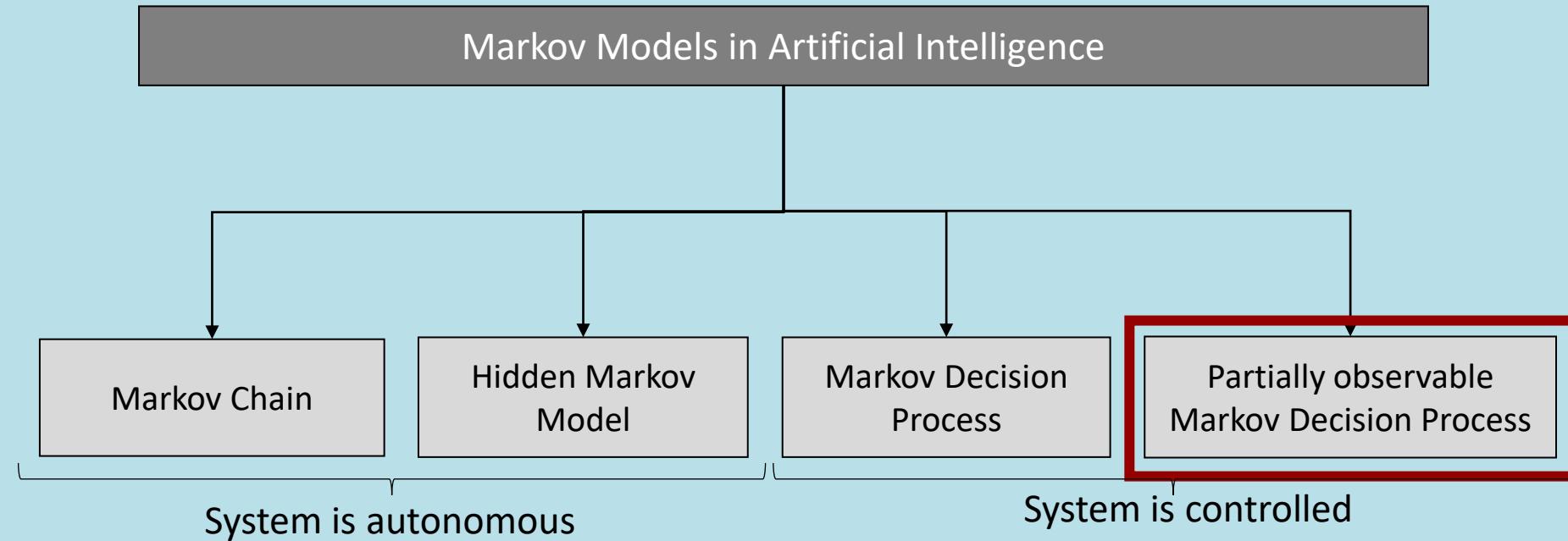
- Start with some initial policy  $\pi_0$  and alternate between the following steps:
  - **Policy evaluation:** calculate  $U^{\pi_i}(s)$  for every state  $s$
  - **Policy improvement:** calculate a new policy  $\pi_{i+1}$  based on the updated utilities

$$\pi^{i+1}(s) = \arg \max_{a \in A(s)} \sum_{s'} P(s'| s, a) U^{\pi_i}(s')$$

## 8.5 Mixing things up

- Do not need to update every state every time
  - Makes sense to focus on states where we will spend most of our time
- In policy iteration, may not make sense to compute state values exactly
  - Will soon change policy anyway
  - Just use some value iteration updates (with fixed policy, as we did earlier)
- Being flexible leads to faster solutions

## 8.5 Types of Markov Models – Controlled Markov Models



- Markov process + partial observability = HMM
  - Markov process + actions = MDP
  - Markov process + partial observability + actions = HMM + actions = MDP + partial observability = POMDP

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

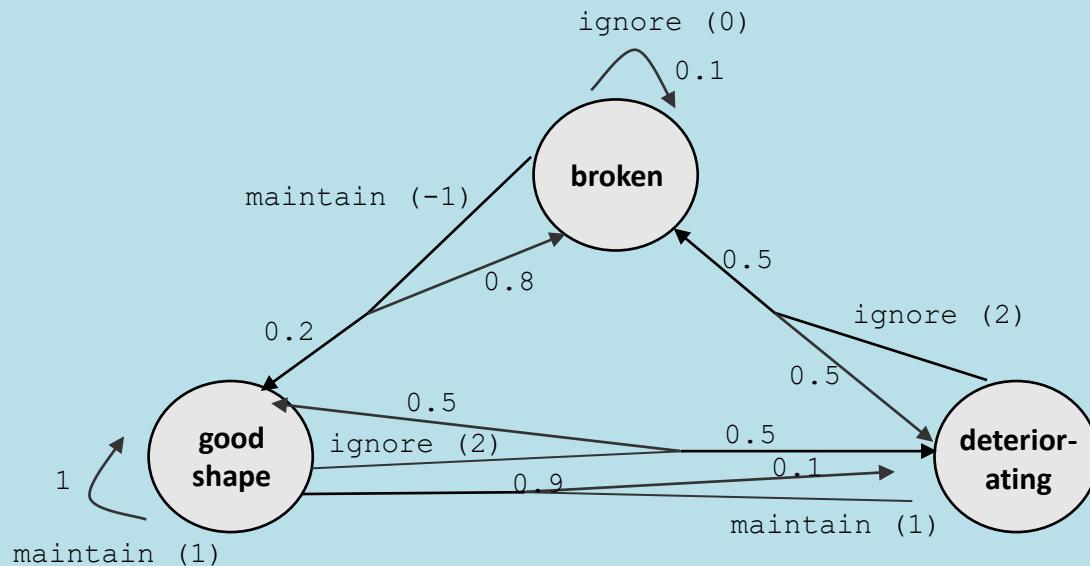
## 8.5 Example 2: Car Part Failures as MDP

- Car part (e.g. V-belt) can be in one of three states

$$S = \{good, deteriorating, broken\}$$

- Can take two actions:

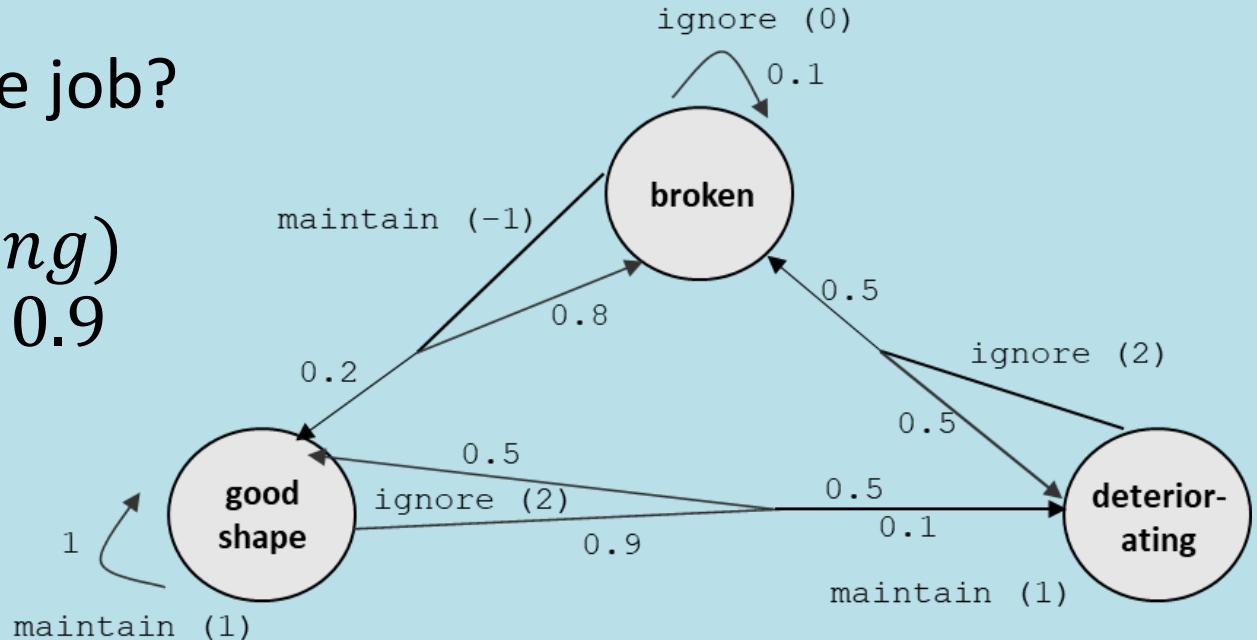
$$A = \{maintain, ignore\}$$



Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.5 Example 2: Car Part Failures as POMDP

- Need to specify observations
- E.g., does part fail on a single job?  
 $P(\text{fail} \mid \text{good shape}) = 0.1, P(\text{fail} \mid \text{deteriorating}) = 0.2, P(\text{fail} \mid \text{broken}) = 0.9$



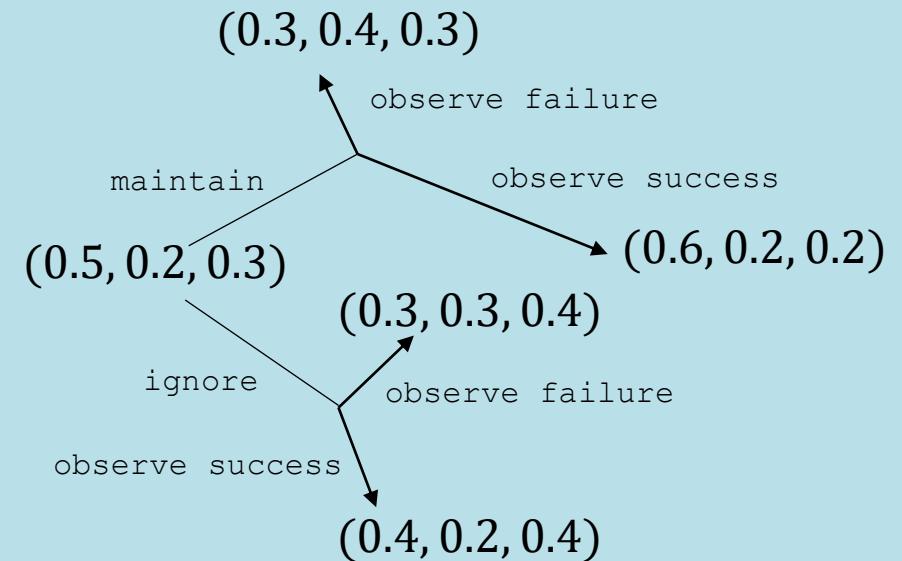
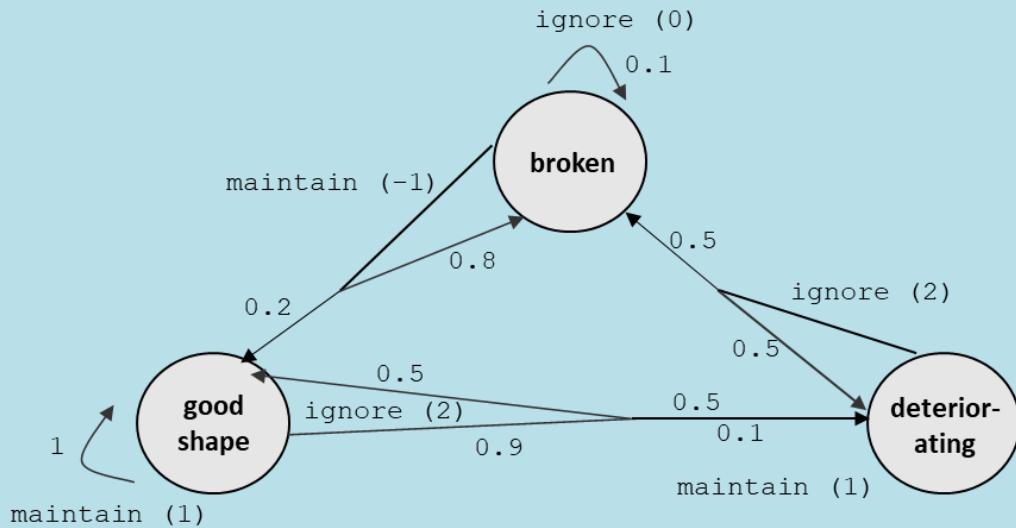
Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.5 Definition of Partially observable Markov Decision Process

- Cannot simply use  $\pi(s)$  because we do not know  $s$
- **Simple idea:** Use a sensormodel  $P(e | s)$
- **Other idea:** We can maintain a probability distribution over  $s$  using filtering:  
$$P(S_t | A_1 = a_1, O_1 = o_1, \dots, A_{t-1} = a_{t-1}, O_{t-1} = o_{t-1})$$
- This gives a belief state  $b$  where  $b(s)$  is our current probability for  $s$
- **Key observation:** policy only needs to depend on  $b, \pi(b)$

## 8.5 Solving a POMDP as an MDP on Belief States

- If we think of the belief state as the state, then the state is observable and we have an MDP



- New beliefstate  $b'(s') = \alpha P(e | s') \sum_s P(s'|s, a)b(s)$

Adapted from Russell, S., & Norvig, P. (2016); Waldmann, K. H. & Stocker, U. M. (2012)

## 8.5 Acting on Belief States

- For belief state  $b$  do action  $a = \pi^*(b)$
- Perceive  $e$
- Compute

$$FORWARD(b, a, e) = b'(s') = \alpha P(e | s') \sum_s P(s' | s, a) b(s)$$

- Solve with value iteration
- **Problem:** very complex problem, algorithm for MDP are inefficient
- **Alternative:** Model transitions and perceptions as dynamic Bayesian networks

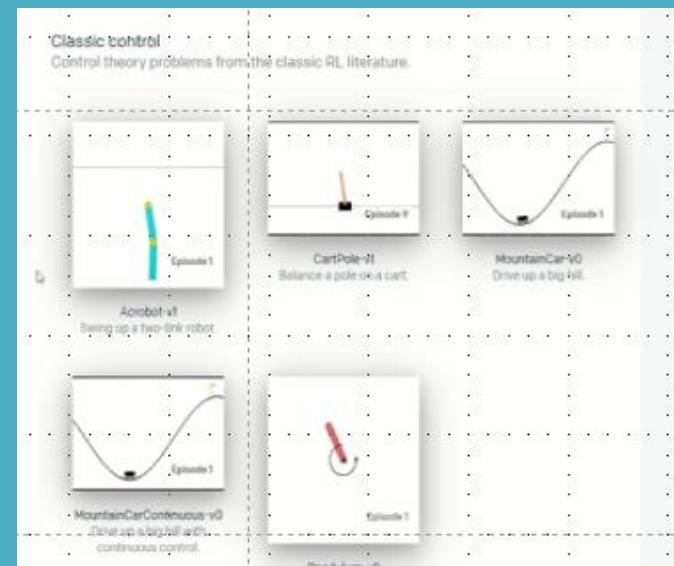
# Your turn!

### Task

Please explain in your own words

- The differences between “Partially observable Markov Decision Process” and “Hidden Markov Models”
- What is the Belmann equitation, for what is it good for?

# Your next project: Intelligent Agents in Action



# 8. Exercises

## Workbook Exercises

- Please read the chapters 13 to 17 from Rusell, S., & Norvig, P. (2016) to understand the theory behind the concepts of this lecture. Then work through the exercises of each chapter.

## Coding Exercises

*The following coding exercise is an old bonus task for students.*

- Implement a markov chain in python to solve the example we solved by hand
- Generalize your markov chain code in a module you can use for other types of problems



Previous  
Bonus Task!

## 8. References

### Literature

1. Frochte, J. (2020). *Maschinelles Lernen: Grundlagen und Algorithmen in Python*.
2. Heckerman, D. E., Horvitz, E. J., & Nathwani, B. N. (1992). Toward normative expert systems: Part I - the pathfinder project. *Methods of information in medicine*, 31(02), 90-105.
3. Gigerenzer, G., & Selten, R. (Eds.). (2002). *Bounded rationality: The adaptive toolbox*. MIT press.
4. Jung, D. (2019): Robo-Advisory and Decision Inertia - Experimental Studies of Human Behaviour in Economic Decision-Making. Dissertation. Online available at: <https://publikationen.bibliothek.kit.edu/1000090689>
5. Jung, D., Dorner, V., Glaser, F., & Morana, S. (2018). Robo-advisory. *Business & Information Systems Engineering*, 60(1), 81-86.
6. Kahneman, D. (2012). *Schnelles Denken, langsames Denken*. Siedler Verlag.
7. Messerli, F. H., Sarmadi, B., Aminuddin, F., Hamid, M., Saari, N., Abdul-Hamid, A., & Ismail, A. (2012). Chocolate and Your Health. *N Engl J Med*, 367(16), 1562-4.
8. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (pp. 841-848). Online available at: <http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf>
9. Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Global Edition.
10. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721), 523-529.

## 8. References

10. Viterbi, A. J. (1967). "Error bounds for convolution codes and an asymptotically optimal decoding algorithm." IEEE Transactions on Information Theory IT-13: 260-269.
11. Waldmann, K. H., & Stocker, U. M. (2012). Stochastische Modelle: Eine anwendungsorientierte Einführung. Springer-Verlag.

### Images

*All images that were not marked other ways are made by myself, or licensed ↗CC0 from ↗Pixabay.*

### Further reading

- Probabilistic reasoning and decision theory and cognitive science has huge influence on artificial intelligence. From the past to the present cognitive scientists and decision theories like Herbert Simon in the beginning or Geoffrey Hinton today are responsible for pathbreaking findings in AI. Take a look at their work to get a non-computer-science perspective on many AI topics and problems.
- Bounded rationality and decision theory influences your everyday decision-making more than you think (see Daniel Kahnemann's [↗Thinking fast and slow](#), and Richard Thaler's [↗Nudging](#), both Nobel-Price-Lauratees).
- Robo-advisors are a very new type of information systems that target to help humans to overcome their cognitive limitations with decision theory and artificial intelligence and decision-making. If you are interested you can take a look at my PHD research ([↗Literature Overview](#))

## 8. Glossary

<b>Bayes Rule</b>	<i>Describes the probability of an event, based on prior knowledge of conditions that might be related to the event.</i>
<b>Bayesian Networks</b>	<i>Represents a set of variables and their conditional dependencies via a directed acyclic graph</i>
<b>De Finetti's theorem</b>	<i>De Finetti's theorem implies that no rational agent can have beliefs that violate the axioms of probability.</i>
<b>Decision Theory</b>	<i>Decision Theory is a combination of probability and utility theory to analyse and model decision-making. Utility theory is used to represent and infer preferences, while probability theory allows us to analyze chance events in a logically sound manner.</i>
<b>Decision-Theoretic Agent</b>	<i>A decision-theoretic agent that selects rational actions based on decision and utility theory.</i>
<b>Random Variables</b>	<i>A Random Variable is a variable whose possible values are numerical outcomes of a random phenomenon</i>
<b>Markov Property</b>	<i>if you want to predict where the chain will be at a future time, and if we know the present state, then the entire past history is irrelevant</i>
<b>Maximizing Expected Utility principle (MEU)</b>	<i>A rational agent should choose the action that maximizes agent's expected utility</i>
<b>Statistical Independence</b>	<i>Two events are independent, statistically independent if the occurrence of one does not affect the probability of occurrence of the other</i>
<b>Robo-advisor</b>	<i>Robo-advisors are information systems that computes different investment recommendations and supports individual portfolio selection based on State-of-the-Art artificial intelligence and decision theory</i>