

ADA511: Data science and data-driven engineering

Steffen Mæland
PierGianLuca Porta Mana

2023-06-24

Table of contents

Preface

Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house. (H. Poincaré)

****WARNING: THIS IS A WORKING DRAFT. TEXT WILL CHANGE A LOT. MANY PASSAGES ARE JUST TEMPORARY, INCOHERENT, AND DISJOINTED.**

To be written.

- Difference between car mechanic and automotive engineer
- “Engineering based on data” is just how engineering and science in general have been in the past 400 years or so. Nothing new there.
- The amount of available data has changed. This may lead to a reduction – or in some cases an increase – in uncertainty, and therefore to different solutions.
- Luckily the fundamental theory to deal with large amount of data is exactly the same to deal with small amounts. So the foundations haven’t changed.

This course makes you acquainted with the foundations.

Part I

An invitation

1 Accept or discard?

Let's start with a question that could arise in a particular engineering problem:

A particular kind of electronic component is produced on an assembly line. At the end of the line, there is an automated inspection device that works as follows with every newly produced component coming out of the line.

The inspection device first makes some tests on the new component. The tests give an uncertain forecast of whether that component will fail within its first year of use, or after.

Then the device decides whether the component is accepted and packaged for sale, or discarded and thrown away.

When a new electronic component is sold, the manufacturer has a net gain of 1\$. If the component fails within a year of use, however, the manufacturer incur net *loss* of 11\$ (12\$ loss, minus the 1\$ gained at first), owing to warranty refunds and damage costs to be paid to the buyer. When a new electronic component is discarded, the manufacturer has 0\$ net gain.

For a specific new electronic component, just come out of the assembly line, the tests of the automated inspection device indicate that there is a 10% probability that the component will fail within its first year of use.

Should the inspection device accept or discard the new component?

First, try to give and motivate an answer.



This is not the real question of this exercise, however. In fact it doesn't matter if you don't get the correct answer; not even if you don't manage to get an answer at all.



Very first exercise!

The purpose here is for you to do some introspection about your own reasoning. Then examine and discuss these points:

- Which numerical elements in the problem seem to affect the answer?
- Can these numerical elements be clearly separated? How would you separate them?
- How would the answer change, if these numerical elements were changed? Feel free to change them, also in extreme ways, and see how the answer would change.
- Could we solve the problem if we didn't have the probabilities? Why?
- Could we solve the problem if we didn't know the various gains and losses? Why?
- Can this problem be somehow abstracted, and then transformed into another one with completely different details? For instance, consider translating along these lines:
 - inspection device → computer pilot of self-driving car
 - tests → camera image
 - fail within a year → pedestrian in front of car
 - accept/discard → keep on going/ break

2 Framework

2.1 What does the intro problem tell us?

Let's approach the "accept or discard?" problem of the previous chapter ?? in an intuitive way.

First let's say that we **accept** the component. What happens?

We must try to make sense of that 10% probability that the component fails within a year. Different people do this with different imagination tricks. We can imagine, for instance, that this situation is repeated 100 times. In 10 of these repetitions the accepted electronic component is sold and fails within a year after selling. In the remaining 90 repetitions, the component is sold and works fine for at least a year.

In each of the 10 imaginary repetitions in which the component fails early, the manufacturer loses 11\$. That's a total loss of $10 \cdot 11\$ = 110\$$. In each of the 90 imaginary repetitions in which the component doesn't fail early, the manufacturer gains 1\$. That's a total gain of 90\$. So over all 100 imaginary repetitions the manufacturer gains

$$10 \cdot (-11\$) + 90 \cdot 1\$ = -20\$,$$

that is, the manufacturer has not gained, but *lost* 20\$! That's an average of 0.2\$ *lost* per repetition.

Now let's say that we **discard** the component instead. What happens? In this case we don't need to invoke imaginary repetitions, but even if we do, it's clear that the manufacturer doesn't gain or lose anything – that is, the "gain" is 0\$ – in each and all of the repetitions.

The conclusion is that if in a situation like this we accept the component, then we'll lose 0.2\$ on average; whereas if we discard it, then on average we won't lose anything or gain anything.

We're jumping the gun here, because we haven't learned the method to solve this problem yet!

Obviously the best, or “least worst”, decision to make is to **discard** the component.

Exercises

1. Now that we have an idea of the general reasoning, check what happens with different values of the probability of failure and of the failure cost: is it still best to discard? For instance, try with
 - failure probability 10% and failure cost 5\$;
 - failure probability 5% and failure cost 11\$;
 - failure probability 10%, failure cost 11\$, non-failure gain 2\$.

Feel free to get wild and do plots.

2. Identify the failure probability at which accepting the component doesn't lead to any loss or any gain, so it doesn't matter whether we discard or accept. (You can solve this as you prefer: analytically with an equation, visually with a plot, by trial & error on several cases, or whatnot.)
3. Consider the special case with failure probability 0% and failure cost 10\$. This means no new component will ever fail. To decide in such a case we do not need imaginary repetitions; but **confirm** that we arrive at the same logical conclusion whether we reason through imaginary repetitions or not.
4. Consider this completely different problem:

A patient is examined by a brand-new medical diagnostics AI system.

The AI first performs some clinical tests on the patient. The tests give an uncertain forecast of whether the patient has a particular disease or not.

Then the AI decides whether the patient should be dismissed without treatment, or treated with a particular medicine.

If the patient is dismissed, then the life expectancy doesn't increase or decrease

if the disease is not present, but it decreases by 10 years if the disease is actually present. If the patient is treated, then the life expectancy decreases by 1 year if the disease is not present (owing to treatment side-effects), but also if the disease is present (because it cures the disease, so the life expectancy doesn't decrease by 10 years; but it still decreases by 1 year owing to the side effects).

For this patient, the clinical tests indicate that there is a 10% probability that the patient has the disease.

Should the diagnostic AI dismiss or treat the patient? Find differences and similarities, even numerical, with the assembly-line problem.

From the solution of the problem and from the exploring exercises, we gather some instructive points:

- Is it enough if we simply know that the component is less likely to fail than not? in other words, if we simply know that the probability of failure is less than 50%?

Obviously not. We found that if the failure probability is 10% then it's best to discard; but if it's 5% then it's best to accept. In both cases the component was less likely to fail than not, but the decisions were different. Moreover, we found that the probability affected the loss if one made the non-optimal decision. Therefore:

Knowledge of exact probabilities is absolutely necessary for making the best decision

- Is it enough if we simply know that failure leads to a cost? that is, that its gain is less than the gain for non-failure?

Obviously not. The situation is similar to that with the probability. In the exercise we found that if the failure cost is 11\$ then it's best to discard; but if it's 5\$ then

it's best to accept. It's also best to accept if the failure cost is 11\$ but the non-failure gain is 2\$. Therefore:

Knowledge of the exact gains and losses is absolutely necessary for making the best decision

- Is this kind of decision situation only relevant to assembly lines and sales?

By all means not. We found a clinical situation that's exactly analogous: there's uncertainty, there are gains and losses (of time rather than money), and the best decision depends on both.

2.2 Our focus: decision-making, inference, and data science

Every data-driven engineering project is unique, with its unique difficulties and problems. But there are also problems common to all engineering projects.

In the scenarios we explored above, we found an extremely important problem-pattern. There is a decision or choice to make (and “not deciding” is not an option – or it's just another kind choice). Making a particular decision will lead to some consequences, some leading to a desired goal, others leading to something undesirable. The decision is difficult because its consequences are not known with certainty, given the information and data available in the problem. We may lack information and data about past or present details, about future events and responses, and so on. This is what we call a problem of **decision-making under uncertainty** or **under risk**¹, or simply a “decision problem” for short.

This problem-pattern appears literally everywhere. But our explored scenarios also suggest that this problem-pattern has a sort of systematic solution method.

In this course we're going to focus on decision problems and their systematic solution method. We'll learn a framework and some abstract notions that allow us to frame and analyse this kind of problem, and we'll learn a universal set of

¹We'll avoid the word “risk” because it has several different technical meanings in the literature, some even contradictory.

principles to solve it. This set of principles goes under the name of **Decision Theory**.

But what do decision-making under uncertainty and Decision Theory have to do with *data* and *data science*? The three are profoundly and tightly related on many different planes:

- We saw that *probability* values are essential in a decision problem. How do we find them? As you can imagine, *data* play an important part in their calculation. In our intro example, the failure probability must come from observations or experiments on similar electronic components.
- We saw that also the values of *gains and losses* are essential. *Data* play an important part in their calculation as well.
- *Data science* is based on the laws of *Decision Theory*. Here's an analogy: a rocket engineer relies on fundamental physical laws (balance of momentum, energy, and so on) for making a rocket work. Failure to account for those laws leads at best to sub-optimal solutions, at worst to disasters. As we shall see, the same is true for a data scientist and the rules of decision theory.
- *Machine-learning* algorithms, in particular, are realizations or approximations of the rules of *Decision Theory*. This is clear, for instance, considering that the main task of a machine-learning classifier is to decide among possible output labels or classes.
- The rules of *Decision Theory* are also the foundations upon which *artificial-intelligence* agents, which must make optimal inferences and decisions, are built.

These five planes will constitute the major parts of the present course.

@@ TODO add examples: algorithm giving outputs is a decision agent. @@ Include one with <https://hjerterisiko.helse.direktoratet.no>

🔗 For the extra curious

Decision theory in expert systems and artificial intelligence

There are other important aspects in engineering problems, besides the one of making decisions under uncertainty. For instance the *discovery* or the *invention* of new technologies and solutions. These aspects can barely be planned or decided; but their fruits, once available, should be handled and used optimally – thus leading to a decision problem.

Artificial intelligence is proving to be a valuable aid in these more creative aspects too. This kind of use of AI is outside the scope of the present notes. Some aspects of this creativity-assisting use, however, do fall within the domain of the present notes. A pattern-searching algorithm, for example, can be optimized by means of the method we are going to study.

2.3 Our goal: optimality, not “success”

What should we demand from a systematic method for solving decision problems?

By definition, in a decision problem under uncertainty there is generally no method to *determine* the decision that surely leads to the desired consequence – if such a method existed, then the problem would not have any uncertainty! Therefore, if there is a method to deal with decision problems, its goal cannot be the determination of the *successful* decision. This also means that a priori we cannot blame an engineer for making an unsuccessful decision in a situation of uncertainty.

Imagine two persons, Henry and Tina, who must bet on “heads” or “tails” under the following conditions (but who otherwise don’t get any special thrill from betting):

- If the bet is “heads” and the coin lands “heads”, the person wins a *small* amount of money; but if it lands “tails”, they lose a *large* amount of money.
- If the bet is “tails” and the coin lands “tails”, the person *wins* a small amount of money; if it lands “heads”, they lose the same *small* amount of money.

Henry chooses the first bet, on “heads”. Tina chooses the second bet, on “tails”. The coin comes down “heads”. So Henry wins the small amount of money, while Tina loses the same small amount. What would we say about their decisions?

Henry's decision was lucky, and yet *irrational*: he risked losing much more money than in the second bet, without any possibility of at least winning more. Tina's decision was unlucky, and yet *rational*: the possibility and amount of winning was the same in the two bets, and she chose the bet with the least amount of loss. We expect that any person making Henry's decision in similar, future bets will eventually lose more money than any person making Tina's decision.

This example shows two points. First, "success" is generally not a good criterion to judge a decision under uncertainty; success can be the pure outcome of luck, not of smarts. Second, even if there is no method to determine which decision is successful, there is a method to determine which decision is rational or **optimal**, given the particular gains, losses, and uncertainties involved in the decision problem. We had a glimpse of this method in our introductory scenarios.

Let us emphasize, however, that we are not giving up on "success", or trading it for "optimality". Indeed we'll find that **Decision Theory automatically leads to the *successful* decision** in problems where uncertainty is not present or is irrelevant. It's a win-win. It's important to keep this point in mind:

Aiming to find the solutions that are *successful* can make us *fail* to find those that are optimal when the successful ones cannot be determined.
Aiming to find the solutions that are *optimal* makes us automatically find those that are *successful* when those can be determined.

We shall later witness this fact with our own eyes, and will take it up again in the discussion of some misleading techniques to evaluate machine-learning algorithms.

2.4 Decision Theory

So far we have mentioned that Decision Theory has the following features:

- ✓ it tells us what's optimal and, when possible, what's successful

- ✓ it takes into consideration decisions, consequences, costs and gains
- ✓ it is able to deal with uncertainties

What other kinds of features should we demand from it, in order to be applied to as many kinds of decision problems as possible, and to be relevant for data science?

If we find an optimal decision in regards to some outcome, it may still happen that the decision can be realized in several ways that are equivalent in regard to the outcome, but inequivalent in regard to time or resources. In the assembly-line scenario, for example, the decision **discard** could be carried out by burning, recycling, and so on. We thus face a decision within a decision. In general, a decision problem may involve several decision sub-problems, in turn involving decision sub-sub-problems, and so on.

In data science, a common engineering goal is to design and build an automated or AI-based device capable of making an optimal decision in a specific kind of uncertain situations. Think for instance of an aeronautic engineer designing an autopilot system, or a software company designing an image classifier.

Decision Theory turns out to meet these demands too, thanks to the following features:

- ✓ it is susceptible to recursive, sequential, and modular application
- ✓ it can be used not only for human decision-makers, but also for automated or AI devices

Decision Theory has a long history, going back to Leibniz in the 1600s and partly even to Aristotle in the —300s, and appearing in its present form around 1920–1960. What’s remarkable about it is that it is not only *a* framework, but *the* framework we must use. A logico-mathematical theorem shows that **any framework that does not break basic optimality and rationality criteria has to be equivalent to Decision Theory**. In other words, any “alternative” framework may use different technical terminology and

rewrite mathematical operations in a different way, but it boils down to the same notions and operations of Decision Theory. So if you wanted to invent and use another framework, then either (a) it would lead to some irrational or illogical consequences, or (b) it would lead to results identical to Decision Theory's. Many frameworks that you are probably familiar with, such as optimization theory or Boolean logic, are just specific applications or particular cases of Decision Theory.

Thus we list one more important characteristic of Decision Theory:

- ✓ it is **normative**

Normative contrasts with *descriptive*. The purpose of Decision Theory is not to describe, for example, how human decision-makers typically make decisions. Because human decision-makers typically make irrational, sub-optimal, or biased decisions. That's exactly what we want to avoid and improve!

🔑 For the extra curious

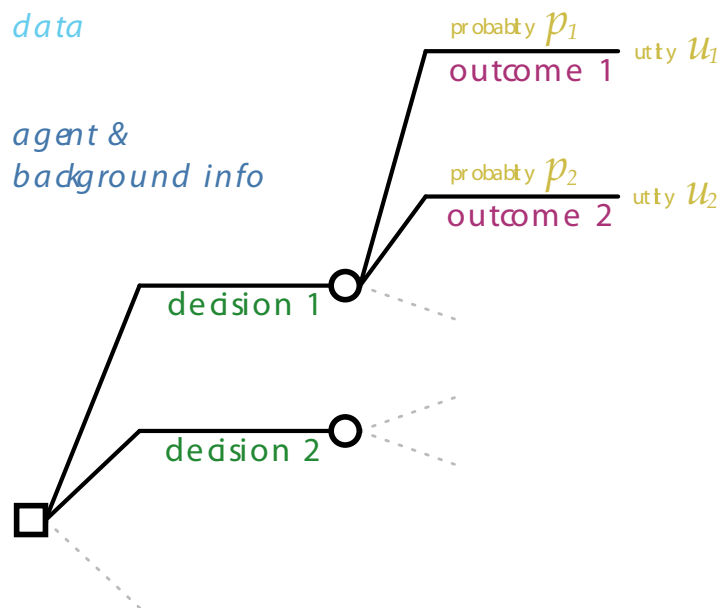
- *Judgment under uncertainty*
- *Heuristics and Biases*
- *Thinking, Fast and Slow*

3 Basic decision problems

Decision Theory analyses any decision-making problem in terms of nested or sequential *basic* or *minimal* decision problems. The assembly-line scenario of the introduction ?? is an example.







3.1 Graphical representation and elements

A basic decision problem can be represented by a diagram like this:



It has one *decision node*, usually represented by a square \blacksquare , from which the available decisions depart as lines. Each decision leads to an *uncertainty node*, usually represented by a circle \bullet , from which the possible outcomes depart as lines. Each outcome leads to a particular utility value. The uncertainty of each outcome is quantified by a probability.

A basic decision problem is analysed in terms of these elements:

-  **Agent**, and **background** or **prior information**. The agent is the person or device that has to make the decision. An agent always possess (or has been programmed with) specific background information that is used and taken for granted in the decision-making process. This background information determines the probabilities and utilities of the outcomes, together with other available data and information. Since different agents typically have different background information, we shall somehow conflate agents and prior information.
-  **Decisions**, also called **courses of actions**, available to the agent. They are assumed to be mutually exclusive and exhaustive; this can always be achieved by recombining them if necessary, as we'll discuss later.
-  **Outcomes** of the possible decisions. Every decision can have a different set of outcomes, or some outcomes can appear for several or all decisions (in this case they are reported multiple times in the decision diagram). Note that even if an outcome can happen for two or more different decisions, its probabilities can still be different depending on the decision.
-  **Probabilities** for each of the outcomes. Their values typically depend on the background information, the decision, and the additional data.
-  **Utilities**: the gains or losses associated with each of the possible outcomes. Their values also depend on the background information, the decision, and the additional data.
-  **Data** and other **additional information**, sometimes called **evidence**. They differ from the background information in that they can change with every decision instance made by the same agent, while the background information stays the same. In the assembly-line scenario, for example, the test results could be different for every new electric component.

We'll use the neutral pronouns *it/its* when referring to an agent, since an agent could be a person or a machine.

Note that it is not always the case that the *outcomes* are unknown and the *data* are known. As we'll discuss later, in some situations we reason in hypothetical or counterfactual ways, using hypothetical data and considering outcomes which have already occurred.

Reading

§ 1.1.4 in *Artificial Intelligence*

Exercise

- Identify the elements above in the assembly-line decision problem of the introduction ??.
- Sketch the diagram of the assembly-line decision problem.

Some of the decision-problem elements listed above may need to be in turn analysed by a decision sub-problem. For instance, the utilities could depend on uncertain factors: thus we have a decision sub-problem to determine the optimal values to be used for the utilities of the main problem. This is an example of the modular character of decision theory.

We shall soon see how to mathematically represent these elements.

The elements above must be identified unambiguously in every decision problem. The analysis into these elements greatly helps in making the problem and its solution well-defined.

An advantage of decision theory is that its application *forces* us to make sense of an engineering problem. A useful procedure is to formulate the general problem in terms of the elements above, identifying them clearly. If the definition of any of the terms involves uncertainty of further decisions, then we analyse it in turn as a decision sub-problem, and so on.


Suppose someone (probably a politician) says: “We must solve the energy crisis by reducing energy consumption or producing more energy”. From a decision-making point of view, this person has effectively said *nothing whatsoever*.

Remember: What matters is to be able to identify these elements in a concrete problem, understanding their role. Their technical names don't matter.

By definition the “energy crisis” is the problem that energy production doesn’t meet demand. So this person has only said “we would like the problem to be solved”, without specifying any solution. A decision-theory approach to this problem requires us to specify which concrete courses of action should be taken for reducing consumption or increasing productions, and what their probable outcomes, costs, and gains would be.


3.2 Inference, utility, maximization

The solution of a basic decision-making problem can be roughly divided into three main stages: inference, utility assessment, and expected-utility maximization.

 **Inference** is the stage where the probabilities of the possible outcomes are calculated. Its rules are given by the **Probability Calculus**. Inference is independent from decision: in some situations we may simply wish to assess whether some hypotheses, conjectures, or outcomes are more or less plausible than others, without making any decision. This kind of assessment can be very important in problems of communication and storage, and it is specially considered by **Information Theory**.


The calculation of probabilities can be the part that demands most thinking, time, and computational resources in a decision problem. It is also the part that typically makes most use of data – and where data can be most easily misused.

Roughly half of this course will be devoted in understanding the laws of inference, their applications, uses, and misuses.

 **Utility assesment** is the stage where the gains or losses of the possible outcomes are calculated. Often this stage requires further inferences and further decision-making sub-problems. The theory underlying utility assessment is still much underdeveloped, compared to probability theory.

 For the extra curious

See MacKay’s options-vs-costs rational analysis in [Sustainable Energy – without the hot air](#)

 **Expected-utility maximization** is the final stage where the probabilities and gains or costs of the possible outcomes are combined, in order to determine the optimal decision.

Part II

Inference

4 What is an inference?

In the assembly-line decision problem of § ??, the probability of early failure (and that of late failure), in view of the test results, was very important in determining the optimal decision. If the probability had been 5% instead of 10%, the optimal decision would have been different. In that scenario the probabilities of the outcomes in view of the test results were already given. In real decision problems, however, probabilities almost always need to be calculated, and their calculation can be the most time- and resource-demanding step in solving a decision problem.


We'll loosely refer to problems of calculating probabilities as “*inference* problems”, and to their calculation as “drawing an inference”. Drawing inferences is very often a goal or need in itself, with no underlying decision process.


4.1 The wide scope and characteristics of inferences

Let's see a couple more informal examples of inference problems. For some of them an underlying decision problem is also alluded to:

- A. Looking at the weather we try to assess if it'll rain today, to decide whether to take an umbrella.
- B. Considering a patient's symptoms, test results, and medical history, a clinician tries to assess which disease affects a patient, so as to decide on the optimal treatment.



- C. Looking at the present game position  the X-player, which moves next, wonders whether placing the next X on the mid-right position leads to a win.

- D. From the current set of camera frames, the computer of a self-driving car needs to assess whether a particular patch of colours in the frames is a person, so as to slow down the car and stop.
- E. Given that $G = 6.67 \cdot 10^{-11} \text{ m}^3 \text{ s}^{-2} \text{ kg}^{-1}$, $M = 5.97 \cdot 10^{24} \text{ kg}$ (mass of the Earth), and $r = 6.37 \cdot 10^6 \text{ m}$ (radius of the Earth), [a rocket engineer needs to know](#) how much is $\sqrt{2GM/r}$.
- F. We'd like to know whether the rolled die is going to show .
- G. An [aircraft's autopilot system](#) needs to assess how much the aircraft's [roll](#) will change if the right wing's [angle of attack](#) is increased by 0.1 rad.
- H. By looking at the dimensions, shape, texture of a newly dug-out fossil bone, an archaeologist wonders whether it belonged to a Tyrannosaurus rex.
- I. A voltage test on a newly produced electronic component yields a reading of 100 mV. The electronic component turns out to be defective. An engineer wants to assess whether the voltage-test reading could have been 100 mV, if the component had not been defective.
- J. Same as above, but the engineer wants to assess whether the voltage-test reading could have been 80 mV, if the component had not been defective.
- K. From measurements of the Sun's energy output and of concentrations of various substances in the Earth's atmosphere over the past 500 000 years, and of the emission rates of various substances in the years 1900–2022, climatologists and geophysicists try to assess the rate of mean-temperature increase in the years 2023–2100.

 For the extra curious

Ch. 10 in [A Survival Guide to the Misinformation Age](#).

Exercises

5. For each example above, pinpoint what has to be inferred, and also the *agent* interested in the inference.
6. Point out which of the examples above *explicitly* give data or information that should be used for the inference.
7. For the examples that do not give explicit data or information, speculate what information could be implicitly assumed. For those that do give explicit data, speculate which other additional information could be implicitly assumed.
8. Can any of the inferences above be done perfectly, that is, without any uncertainty, based the data given explicitly or implicitly?
9. Find the examples that explicitly involve a decision. In which of them does the decision affect the results of the inference? In which it does not?
10. Are any of the inferences “*one-time only*” – that is, their object or the data on which they are based have never happened before and will never happen again?
11. Are any of the inferences based on data and information that come chronologically *after* the object of the inference?
12. Are any of the inferences about something that is actually already known to the agent that’s making the inference?
13. Are any of the inferences about something that actually did not happen?
14. Do any of the inferences use “data” or “information” that are actually known (within the scenario itself) to be fictive, that is, *not* real?

From the examples and from your answers to the exercise we observe some very important characteristics of inferences:

- Some inferences can be made exactly, that is, *without uncertainty*: it is possible to say whether the object of the inference is true or false. Other inferences, instead, involve an uncertainty.
- *All inferences are based on some data and information*, which may be explicitly expressed or only implicitly understood.
- An inference can be about something *past*, but based on *present or future* data and information: inferences can show *all sorts of temporal relations*.
- An inference can be *essentially unrepeatable*, because it's about something unrepeatable or based on unrepeatable data and information.
- The data and information on which an inference is based can actually be unknown; that is, they can be only momentarily contemplated as real. Such an inference is said to be based on **hypothetical reasoning**.
- The object of an inference can actually be something already known to be false or not real: the inference tries to assess it in the case that some data or information had been different. Such an inference is said to be based on **counterfactual reasoning**.

4.2 Basic elements of an inference

Let us already introduce the basic mathematical notation for inferences. We have seen that every inference has an “object” (what is to be assessed) and data and information on which it is based. We call **proposal**¹ the object of the inference, and **conditional**² what the inference is based upon. We separate them with a vertical bar³ “|”, which can be pronounced *given* or *conditional on*:

$$[proposal] \mid [conditional]$$

¹Johnson’s (1924) terminology. Keynes (1921) uses “conclusion”. Modern textbooks do not seem to use any specialized term.

²Modern terminology. Other terms used: “evidence”, “premise”, “supposal”.

³Originally a [solidus](#), introduced by Keynes (1921).

Again: *terminology doesn't matter* (although it's useful). What matters is that we understand the properties and uses of these notions.

There are now two important tasks ahead of us. First, we want to introduce a flexible and enough general mathematical representation for the objects and the bases of an inference. Second, we want to know what are the rules for making correct inferences.

5 Sentences

We have seen that an inference involves at the very least two things: the object of the inference (*proposal*), and the data, information, or hypotheses on which the inference is based (*conditional*).

We also observed that wildly different “items” can be the object of an inference or the information on which the inference is based: measurement results, decision outcomes, hypotheses, not-real events, assumptions, data and information of all kinds (for example, images). In fact, such variety in some cases can make it difficult to pinpoint what an inference is about or what it is based on.

Is there a general, flexible, yet precise way of representing all these kinds of “items”?

5.1 The central components of knowledge representation

When speaking of “data”, what comes to mind to many people is basically numbers or collections of numbers. Maybe numbers, then, could be used to represent all the variety of items exemplified above. This option, however, turns out to be too restrictive.

I give you this number: “8”, saying that it is “data”. But what is it about? You, as an agent, can hardly call this number a piece of information, because you have no clue what to do with it. Instead, if I tell you: “[The number of official planets in the solar system is 8](#)”, then we can say that I’ve given you data. So “data” is not just numbers: a number is not “data” unless there’s an additional verbal, non-numeric context accompanying it, even if only implicitly. Sure, we could represent this meta-data information as numbers too; but this move would only shift the problem one level up: we

would need an auxiliary verbal context explaining what the meta-data numbers are about.

Data can, moreover, be completely non-numeric. A clinician saying “The patient has fully recovered from the disease” (we imagine to know who’s the patient and what was the disease) is giving us a piece of information that we could further use, for instance, to make prognoses about other, similar patients. The clinician’s statement surely is “data”, but essentially non-numeric data. Sure, in some situations we can represent it as “1”, while “0” would represent “not recovered”; but the opposite convention could also be used, or the numbers “0.3” and “174”. These numbers have intrinsically nothing to do with the clinician’s “recovery” data.

But the examples above actually reveal the answer to our needs. In the examples we expressed the data by means of *sentences*. Clearly any measurement result, decision outcome, hypothesis, not-real event, assumption, data, and any piece of information can be expressed by a sentence.

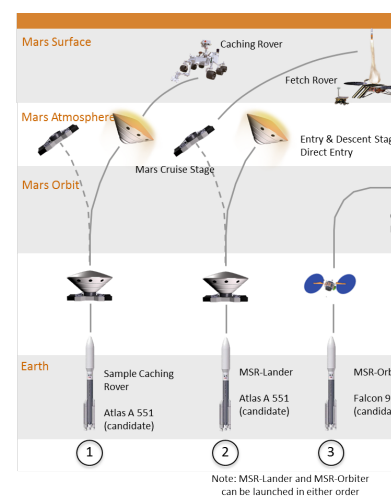
We shall therefore use **sentences**, also called **propositions** or **statements**,¹ to represent and communicate all the kinds of “items” that can be the proposal or conditional of an inference. In some cases we can of course summarize a sentence by a number, as a shorthand, when the full meaning of the sentence is understood.

Sentences are the central components of knowledge representation in AI agents. For example they appear at the heart of automated control programs and fault-management systems in NASA spacecrafts.

Reading

- § 7.1 in *Artificial Intelligence*.
- Take a *quick look* at these:
 - *SMART: A propositional logic-based trade analysis and risk assessment tool for a com-*

¹These three terms are not always equivalent in formal logic, but here we’ll use them as synonyms.



(From the *SMART* paper)

plex mission

- around p.22 in *No More Band-Aids: Integrating FM into the Onboard Execution Architecture*
- §2.1 in *Deliberation for autonomous robots: A survey*
- part IV in *Model-based programming of intelligent embedded systems and robotic space explorers*


5.2 Identifying and working with sentences

But what is a sentence, more exactly? The everyday meaning of this word will work for us, even though there are more precise definitions – and still a lot of research in logic and artificial intelligence on how to define and use sentences. We shall adopt this useful definition:

A “sentence” is a verbal message for which we can determine whether it is **true** or **false**, at least in principle and in such a way that all interested receivers of the message would agree.

For instance, in most engineering contexts the phrase “This valve will operate for at least two months” is a sentence; whereas the phrase “Apples are much tastier than pears” is not, because it’s a matter of personal taste – there’s no objective criterion to determine its truth or falsity (however, the phrase “Rita finds apples tastier than pears” could be a sentence; its truth is found by asking Rita). In a data-science context, the phrase “The neural-network algorithm has better performance than the random-forest one” is *not* a sentence unless we have objectively specified what “*better*” means, for example by using a particular comparison metric.

Some expressions in fact, even involving technical terms, may appear to be sentences at first, but a deeper analysis may reveal that they are not. A famous example is the sentence “The two events (at different spatial locations) are simultaneous”. Einstein showed that there’s no physical way to deter-

 For the extra curious

[Propositions](#)

mine whether such an expression is true or false. Its truth turns out to be a matter of convention (also in Newtonian mechanics). The Theory of Relativity was born from this observation.

One sentence can be expressed by many different phrases and in different languages. For instance, “The temperature is 248.15 K”, “Temperaturen ligger på minus 25 grader”, and “25 °C is the value of the temperature” all represent the *same* sentence.

A sentence can contain numbers, pictures, and graphs.

Working with sentences, and keeping in mind that inference is about sentences, is important in several respects:

First, it leads to **clarity** in engineering problems and makes them more **goal-oriented**. A data engineer must acquire information and convey information. “Acquiring information” does not simply consist in making measurements or counting something: the engineer must understand *what* is being measured and *why*. If data is gathered from third parties, the engineer must ask what exactly the data mean and how they were acquired. In designing and engineering a solution, it is important to understand what information or outcomes the end user exactly wants. The “what”, “why”, “how” are expressed by sentences. A data engineer will often ask “*wait, what do you mean by that?*”. This question is not just an unofficial parenthesis in the official data-transfer workflow between the engineer and someone else. It is an integral part of that workflow: it means that some information has not been completely transferred yet.

Second, it is extremely important in AI and machine-learning design. A (human) engineer may proceed informally when drawing inferences, without worrying about “sentences” unless a need for disambiguation arises. A data engineer who’s *designing* or *programming* an algorithm that will do inferences automatically, must instead be unambiguous and cover beforehand all possible cases that the algorithm will face.

We agree that *the proposal and the conditional of an inference have to be sentences*. This means that the proposal of the inference must be something that can only be true or

🔗 For the extra curious

On the electrodynamics of moving bodies.

false. Many inferences, especially when they concern numerical measurements, are actually collections of inferences. For example, an inference about the result of rolling a die actually consists of six separate inferences with the proposals

‘The result of the roll is 1’
 ‘The result of the roll is 2’
 ...
 ‘The result of the roll is 6’

Later on we shall see how to work with more complex inferences without thinking about this detail. In real applications it can be useful, on some occasions, to pause and reduce an inference to its basic set of **true/false** inferences; this analysis may reveal contradictions in our inference. A simple way to do this is to reduce the complex inference into a set of yes/no questions.

This kind of analysis is also important in information-theoretic situations: the **information content** provided by an inference, when measured in *Shannons*, is related to the minimal amount of yes/no questions that the inference answers.

Exercise

Rewrite each inference scenario of § ?? in a formal way, as one or more inferences

$$[proposal] \mid [conditional]$$

where proposal and conditional are well-defined sentences.

In ambiguous cases, use your judgement and motivate your choices.

5.3 Notation

Writing full sentences would take up *a lot* of space. Even an expression such as “The speed is 10 m/s” is not a sentence, strictly speaking, because it leaves unspecified the speed of what, when it was measured and in which frame of reference,

what we mean by “speed”, how the unit “m/s” is defined, and so on.

Typically we leave the full content of a sentence to be understood from the context, and we denote the sentence by a simple expression such as the one above,

The speed is 10 m/s

or even more compactly introducing physical symbols:

$$v = 10 \text{ m/s}$$

where v is a physical variable denoting the speed; or even writing simply

$$10 \text{ m/s}$$

In some problems it’s useful to introduce symbols to denote sentences. In these notes we’ll use sans-serif italic letters: A, B, a, b, \dots , possibly with sub- or super-scripts. For instance, the sentence “The speed is 10 m/s” could be denoted by the symbol S_{10} . We abbreviate such a definition like this:

$$S_{10} := \text{‘The speed is 10 m/s’}$$

which means “the symbol S_{10} is defined to be the sentence ‘The speed is 10 m/s’”.

! We must be wary of how much we shorten sentences

Consider these three:

‘The speed is measured to be 10 m/s’

‘The speed is set to 10 m/s’

‘The speed is reported, by a third party, to be 10 m/s’

The quantity “10 m/s” is the same in all three sentences, but their meanings are very different. They represent different kinds of data. These differences greatly affect any inference about or from these data. For instance, in the third case an engineer may not take the indirectly-reported speed “10 m/s” at face value, unlike the first case. In a scenario where all three sentences can occur, it would be ambiguous to simply write “ $v = 10 \text{ m/s}$ ”: would the equal-sign mean “measured”, “set”, or “indirectly reported”?

Exercise

How would you denote the three sentences above, to make their differences clear?

5.4 Connecting sentences

5.4.1 Atomic sentences

In analysing the measurement results, decision outcomes, hypotheses, assumptions, data and information that enter into an inference problem, it is convenient to find a collection of **basic sentences** or, using a more technical term, **atomic sentences** out of which all other sentences of interest can be constructed. These atomic sentences often represent elementary pieces of information in the problem.

Consider for instance the following complex sentence, which could appear in our assembly-line scenario:

“The electronic component is still whole after the shock test and the subsequent heating test. The voltage reported in the final power test is either 90 mV or 110 mV.”

In this statement we can identify at least four atomic sentences, which we denote by these symbols:

$s :=$ ‘The component is whole after the shock test’

$h :=$ ‘The component is whole after the heating test’

$v_{90} :=$ ‘The power-test voltage reading is 90 mV’

$v_{110} :=$ ‘The power-test voltage reading is 110 mV’

The inference may actually require additional atomic sentences. For instance, it might become necessary to consider atomic sentences with other values for the reported voltage, such as

$v_{110} :=$ ‘The power-test voltage reading is 100 mV’

$v_{80} :=$ ‘The power-test voltage reading is 80 mV’

and so on.

5.4.2 Connectives

How do we construct complex sentences, like the one above, out of atomic sentences?

We consider three ways: one operation to change a sentence into another related to it, and two operations to combine two or more sentences together. These operations are called **connectives**; you may have encountered them already in Boolean algebra. Our natural language offers many more operations to combine sentences, but these three connectives turn out to be all we need in virtually all engineering and data-science problems:

Not: \neg example:

$\neg s =$ 'The component is broken after the shock test'

And: \wedge example:

$s \wedge h =$ 'The component is whole after the shock and heating tests'

Or: \vee example:

$v_{90} \vee v_{110} =$ 'The power-test voltage reading is 90 mV, or 110 mV, or both'

These connectives can be applied multiple times, to form increasingly complex sentences.

❗ Important subtleties of the connectives:

- There is *no strict correspondence* between the words “not”, “and”, “or” in natural language and the three connectives. For instance the **and** connective could correspond to the words “but” or “whereas”, or just to a comma “,”.
- **Not** means not some kind of complementary quality, but the denial. For instance, \neg ‘The chair is black’ generally does not mean ‘The chair is white’, (although in some situations these two sentences could amount to the same thing).

It’s always best to *declare explicitly what the not*

of a sentence concretely means. In our example we take

\neg ‘The component is whole’ := ‘The component is broken’

But in other examples the negation of “being whole” could comprise several different conditions. A good guideline is to always state the **not** of a sentence in *positive* terms.

- Or does not exclude that both the sentences it connects can be true. So in our example $v_{90} \vee v_{110}$ does not exclude, a priori, that the reported voltage could be both 90 mV and 110 mV. (There is a connective for that: “exclusive-or”, but it can be constructed out of the three we already have.)

From the last remark we see that the sentence

‘The power-test voltage reading is 90 mV or 110 mV’

does *not* correspond to $v_{90} \vee v_{110}$. It is implicitly understood that a voltage reading cannot yield two different values at the same time. Convince yourself that the correct way to write that sentence is this:

$$(v_{90} \vee v_{110}) \wedge \neg(v_{90} \wedge v_{110})$$

Finally, the full complex sentence of the present example can be written in symbols as follows:

“The electronic component is still whole after the shock test and the subsequent heating test. The voltage reported in the final power test is either 90 mV or 110 mV.”

$$s \wedge h \wedge (v_{90} \vee v_{110}) \wedge \neg(v_{90} \wedge v_{110})$$

Reading

Just take a quick look at § 7.4.1 in *Artificial Intelligence* and note the similarities with what we've just learned. In these notes we follow a faster approach leading directly to probability logic.

5.5 “If... then...”

Sentences expressing data and information in natural language also appear connected with *if... then...*. For instance: “If the voltage reading is 200 mV, then the component is defective”. This kind of expression actually indicates that the following inference

‘The component is defective’ | ‘The voltage reading is 200 mV’

is **true**.

This kind of information is very important because it often is the starting point from which to arrive at the final inferences we're interested in. We shall discuss it more in detail in the next sections.

Careful

There is a connective in logic, called “**material conditional**”, which is also often translated as “if... then...”. But it is not the same as the inference relation discussed above. “If... then...” in natural language usually denotes an inference rather than a material conditional. Research is still ongoing on these topics. If you are curious and in for a headache, look over *The logic of conditionals*.

We are now equipped with all the notions and symbolic notation to deal with our next task: learning the rules for drawing correct inferences.

6 Truth inference

6.1 A trivial inference

Consider again the assembly-line scenario of § ??, and suppose that an inspector has the following information about an electric component:

This electric component had an early failure (within a year of use). If an electric component fails early, then at production it didn't pass either the heating test or the shock test. This component passed the shock test.

The inspector wants to assess whether the component did not pass the heating test.

From these data and information given, the conclusion is that the component *for sure* did not pass the heating test. This conclusion is certain, and also trivial. But how did we obtain it? Which rules did we follow to arrive at it from the given data?

Formal logic, with its *deduction systems*, is the huge field that formalizes and makes rigorous the rules that a rational person or an artificial intelligence should use in drawing *sure* inferences like the one above. We'll now get a glimpse of it, as a trampoline for jumping towards more general and *uncertain* inferences.

6.2 Analysis and representation of the problem

First let's analyse our simple problem and represent it with more compact symbols.

We can introduce the following atomic sentences and symbols:

h := ‘The component passed the heating test’
 s := ‘The component passed the shock test’
 f := ‘The component had an early failure’
 I := (all other implicit background information)

The inference that the inspector wants to draw can be compactly written:

$$\neg h \mid s \wedge f \wedge I$$

6.3 Truth-inference rules

Formal logic gives us sets of rules for correctly drawing sure inferences, that is, inferences where the truth or falsity of the proposal can be exactly determined – *when such inferences are possible*. These rules can be formulated in different ways, leading to a [wide variety of deduction systems](#). The picture here on the margin, for instance, shows how a proof of how our inference would look like, using the so-called sequent calculus.

One important point is that deduction systems don’t lead to inferences out of nothing: *in order to draw an inference, we must always start from some other inferences that are already known*. This is one of the central points of logic, which you may have already heard about. It is sometimes stated in terms of *axioms* and *theorems*: in order to prove some theorem, we must always start from some axioms.¹

$$\frac{\frac{D \wedge f \vdash \neg h \vee \neg s \quad s \vdash s}{\neg s \wedge s \vdash}}{D \wedge f \wedge s \vdash \neg h}$$

Figure 6.1: The bottom formula is our conclusion. Each line denotes the application of an inference rule. The two formulae with no line above are our initial, known inference, and a tautology.

¹There actually are “inferences” that can be drawn without requiring other ones; but they are all trivial *tautologies*, such as the conclusion “this component failed early, or it didn’t”. They are of little use in a real problem; but are important nevertheless, because they reflect the core of the inference rules.

We can compactly encode all inference rules in the following way. First, represent **true** by the number 1, and **false** by 0. Second, symbolically write that conclusion C is **true**, given assumptions A , as follows:

$$T(C \mid A) = 1 .$$

or with 0 if it's **false**.

What are the data available to the inspector? One is that the component failed, f ; another is that the component passed the shock test, s . We already included both in the conditional of the inference above. The inspector also has another piece of information, which consists in a *known* inference: “if a component fails early, then it didn’t pass either the heating test or the shock test”. We write the fact that this inference is known as follows:

$$T(\neg h \vee \neg s \mid f \wedge I) = 1$$

- The component can either come from the production line in Oslo, or from the one in Rome.
- If the component is defective, it cannot come from Oslo.
- The component is found to be defective.

The question is: from which production line does the component come from?

The answer is obvious: from the Rome line. But how could we draw this obvious and sure inference? Which rules did we follow? Did we make any hidden assumptions, or use information that wasn’t explicitly mentioned?

Logic is the huge field that formalizes and makes rigorous the rules that a rational person or an artificial intelligence should use in drawing sure inferences. We’ll get a glimpse of it here, as a trampoline for jumping towards the more general inferences that we need in data-driven engineering problems.

6.3.1 Analysis of the problem

Let’s write down the basic sentences that constitute our data and the inferences we want to draw. We identify three basic sentences, which we can represent by these symbols:

- $o :=$ 'The component comes from the Oslo line'
- $r :=$ 'The component comes from the Rome line'
- $d :=$ 'The component is defective'

Obviously the inspector possesses even more information which is implicitly understood. It's clear, for instance, that the component cannot come from both Oslo and Rome. Let's denote this information with

- $I :=$ (a long collection of sentences explaining all other implicitly understood information).

With the sentences above we can express more complex details and hypotheses appearing in the inspector's problem, in particular:

- $o \vee r =$ 'The component comes from either the Oslo line or the Rome line'
- $\neg(o \wedge r) =$ 'The component cannot come from both the Oslo and the Rome lines'
- $\$ \neg o$ 'The component does not come from the Oslo line'\$

6.3.2 Data, assumptions, desired conclusions

The inspector knows for certain the following facts:

- $o \vee r$, 'The component comes from either the Oslo line or the Rome line'
- $\neg(o \wedge r)$, 'The component cannot come from both the Oslo and the Rome lines'
- d , 'The component is defective'
- I , all remaining implicit information

We and them all together:

$$d \wedge (o \vee r) \wedge \neg(o \wedge r) \wedge I .$$

The inspector knows, moreover, this hypothetical consequence:

- $\neg o | d \wedge (o \vee r) \wedge \neg(o \wedge r) \wedge I$, if the component is defective, it cannot come from the Oslo production line.
-