ADA511: Data science and data-driven engineering

Steffen Mæland
PierGianLuca Porta Mana

2023-06-12

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.

1 Introduction

To be written: motivation and structure of this course.

2 Framework

Every data-driven engineering problem is unique. But there are also similarities among all engineering problems. We shall now learn a framework that allow us to frame and face any data-driven engineering problem, and any sub-problems into which a problem can be broken down. This framework is build on some notions that help us analyse the problem, and on a set of principles. This set of principles is important because it mathematically guarantees an optimal solution to the problem – within the goals, means, and data into which we framed the problem.

2.1 Decision Theory

There is always a goal underlying any engineering problem. The problem itself is how to reach that goal. Typically there are several possible courses of actions available; the question is which one to choose. Choice of a specific action will lead to some consequences, which could be the goal we want to reach, but could also be something else, possibly undesirable. The decision upon a specific action is often difficult because its consequences are not known with certainty. This uncertainty comes in turn from a more general uncertainty about the whole context of the problem: unknown past or present details, unknown future events and responses, and so on.

@@ example

This is what we call a **decision problem**.

A specific course of action may in turn be realized in several ways that are equivalent in regard to the outcome, but inequivalent in regard to costs, time, resources. We thus face a decision within a decision. In general, a decision problem may involve several decision sub-problems, in turn involving sub-sub-problems, and so on.

The main engineering goal itself could be to design and build an automated device that chooses an optimal course of action in a specific kind of uncertain situation. Think for instance of an aeronautic engineer who is designing an autopilot system.

Therefore, to analyse and tackle this kind of problems we would like to have a framework with the following features: it should take into consideration choices, consequences, costs and gains, and uncertainties; it should be susceptible to recursive application if needed; and it should be suited to being used not only for human engineers, but also for automated or artificial-intelligence devices.

Such a framework does exist: it is called **Decision Theory**.

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, it can 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, are just specific applications or particular cases of Decision Theory.

Decision theory consists of two main theories: **Probability Theory**, which deals with data, information, uncertainty, inference; and **Utility Theory**, which deals with actions, consequences, gain and loss, decisions.

We shall get acquainted with Decision Theory step by step, introducing its main ideas and notions as they become necessary. Let us start with the first building blocks.

2.2 Anatomy of a decision problem

An extremely important – and surprisingly often neglected
– first step in every engineering problem is to define exactly
what the problem is. This means, in particular, to specify
unambiguously the goals, the available data and information,
the available courses of action, the hypotheses of interest.

Decision theory analyses any problem in terms of minimal
decision problems, each of which consists of these elements:

- *Agent*: the person or device that has to make an optimal
  decision.

- *Assumptions*: data and background information that
  are known, or temporarily imagined to be known, to
  the agent.

- *Conclusions*: hypotheses, conjectures, or actual facts
  that the agent wishes to assess; often they are related
  to the unknown consequences of the possible actions.

- *Probabilities* of the conclusions given the assumptions.

- *Courses of actions*: the choices available to the agent.

- *Utilities*: the gains and losses involved in choosing a
  course of action.

Together with the "assumptions" and "conclusions" it is also
useful to keep in mind two more and somewhat distinct elements:

- *Knowns*: the data and information that are actually
  known to the agent.

- *Unknowns*: hypotheses, conjectures, and situations
  whose truth or falsity are actually unknown to the
  agent.

The basic idea is that the agent will estimate effects of the
conclusions given the assumptions, and from this estimate
choose the optimal course of action.

Some of these elements may themselves be open to be analysed by a decision sub-problem. For instance, it may be unknown what the utilities are: thus we have a decision subproblem to determine the optimal values we should choose

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

for them. This is an example of the modularity of decision theory.

The elements above must be unambiguously identified in any decision problem. The analysis of the problem into these elements greatly helps in making the problem and its solution well-defined. Suppose someone (probably a politician) says: "We must solve the energy crisis by reducing energy consumption or producing more energy". This person has effectively said *nothing whatsoever*. 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 giving 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 costs, gains, and probable consequences would be.

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.

There are important aspects of engineering that cannot be captured by decision theory or, likely, by any logico-mathematical framework: the *discovery* or the *invention* of new technologies and solutions, for instance. These aspects cannot be planned or decided; but their fruits, once available, can be handled and used optimally by means of decision theory.

Artificial intelligence is proving to be a valuable aid in these 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 decision theory and of the present notes. A pattern-searching algorithm, for example, can be optimized by means of decision theory, if we have an idea of the kind of patterns that could be useful.

3 First building blocks

We shall first discuss how to represent and work with the *agent*, *assumptions*, *conclusions*, *probability* elements, leaving the *actions* and *utilities* to later.

3.1 Agents

The agent is the person or device that has to make a choice between different courses of action. An agent has a specific set of data and background information available, a specific set of choices, and can incur specific gains or losses dependent on the consequences of the available choices.

It is important to identify the agent or agents involved in a problem, because each one will generally have different data, or different available actions, or different gains and losses. A person buying an insurance policy from an insurance company is an example of decision problem with two agents, the person and the company, that have roughly the same data and a common course of action (buy-sell) that is optimal for both. The optimality comes from the fact that the two agents have very different gains and losses for their various courses of action.

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

> 📖 Reading
>
> § 1.1.4 in *Artificial Intelligence*

3.1.1 Notation

When necessary, agents are typically denoted by capital letters: $A, B, \dots$. But we'll rarely need symbols for them.

3.2 Assumptions & conclusions, knowns & unknowns: Sentences

3.2.1 Assumptions

The "assumptions" often include all or part of the data and information that the agent knows. There is a difference between "assumptions" and "knowns", however. In approaching a decision problem, and especially when considering decision sub-problems of a larger problem, the agent must often reason *hypothetically* or *counterfactually*.

Consider for instance an aeronautics problem where it must be decided whether to replace the fuel currently employed by a particular aircraft model, with a newly produced fuel type. To assess the consequence of employing the new fuel, the engineer must momentarily imagine that the new fuel is actually used and then assess thermodynamic, environmental, and economic consequences of this imagined situation. This is an example of *hypothetical reasoning*. Hypothetical reasoning is sometimes assisted by performing experiments in restricted and controlled conditions, in which the new fuel is really used. Such supporting experiments, however, may not be viable, and in any case they are not full reflections of the hypothetical situation.

The engineer may also have data from another aircraft model with which the new fuel was ultimately *not* used, and may try to assess what the consequences of the new fuel would have been *if it had been replaced* in that model, because such assessment may be easier post-facto. This is an example of *counterfactual reasoning*.

In either instance, the engineer sets up a decision problem in which the assumptions consist of a combination of real data and of a situation that in one case is only imagined, and in the other case never happened. Both kinds of reasoning are staples of scientific research.

3.2.2 Conclusions

The "conclusions" often include all or part of the hypotheses or conjectures whose truths are unknown to the agent,

and the agent would like to assess. Conclusions may also include known data, however. Similarly to "assumptions" and "knowns", there's a difference between "conclusions" and "unknowns".

Consider a data engineer testing the responses of a new machine-learning algorithm, in controlled conditions; this is an example of decision problem about a decision agent. The algorithm being evaluated assesses the truth of a particular situation; this truth is unknown to the algorithm itself, but known to the engineer. More generally, we shall see that an agent may have to assess the truth of a known situation, assuming an unknown one, as a temporary step in a more general inference.

3.2.3 Sentences

Is there a flexible and general way of representing assumptions, conclusions, knowns, unknowns, data, information, hypotheses; and, later, also consequences and actions?

When speaking of "data", what comes to mind to many people is basically numbers or collections of numbers. So numbers could perhaps be used to representing assumptions etc. This option turns out to be too limiting, however.

I give you this number: 8, saying that it is "data". But what is it about? As a decision agent, you 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 some verbal, non-numeric context accompanying it – even if this context is only implicitly understood. Note that representing this meta-data information as numbers only shifts the problem one level up: we would need auxiliary verbal context explaining what the meta-data numbers are about.

Data can, moreover, also 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. In some situations we can represent it as "1", while "0" would represent "not recovered"; yet the opposite convention could also be used, showing that these numbers have really nothing to do with the clinician's data.

But the examples above actually contain the answer to our needs. In them we expressed the data by means of **sentences**. Clearly any piece of information, or hypothesis, conclusion, consequence, action can be expressed by a sentence. We shall therefore use sentences, also called *propositions* or *statements*[1], to represent and communicate assumptions, conclusions, knowns, unknowns, data, information, hypotheses, consequences, and actions. 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.

But what is a sentence? The everyday meaning of this word will work for us, even though there is still a lot of research in logic an 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, at least in principle, whether it is `true` or `false`, in a way that all interested receivers of the message would agree.

📖 For the curious Propositions

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. However, the phrase "Rita finds apples tastier than pears" could be a sentence.

Note that a sentence can contain numbers, and even pictures and graphs: this possibility is not excluded from the definition above.

The use of sentences in our framework has important practical consequences:

---

[1] These three terms are not always equivalent in Formal Logic, but here we'll use them as synonyms.

- **Clarity, analysis, goal-orientation.** A data engineer must acquire information and convey information. Acquiring information is not simply making some measurement 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. 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 the data has not been completely transferred yet.

- **Artificial Intelligence.** Sentences are the central components of knowledge representation and inference in artificial-intelligence agents.

> 📖 Reading
>
> § 7.1 in *Artificial Intelligence*

Notation

We'll denote sentences by sans-serif italic letters: *A*, *B*, *a*, *b*, ... For example,

$$O := \text{'The power output is 100 W'}$$

means that the symbol *O* stands for the sentence above. Often we shall simply write sentences in abbreviated form, when their full meaning is understood from the context; for example "$O = 100\,\text{W}$" or even just "$100\,\text{W}$" for the sentence above.

We'll next see how more complex sentences are built from simpler ones. No matter whether complex or simple, any sentence can be represented by symbols like the ones above.

3.3 Combining sentences

3.3.1 Basic sentences

In analysing the assumptions and conclusions of a decision problem it is convenient to find a collection of **basic sentences**[2] out of which all other sentences of interest can be constructed. Often these basic sentences represent a elementary pieces of information in the problem.

Consider for instance the following data in a High-Performance Computing engineering problem:

'All three CPUs report a temperature of 50 °C.
CPUs #1 and #2 are consuming 100 W, whereas #3 is consuming either 110 W or 90 W.'

We can identify at least seven basic sentences:

- 'CPU #1 reports a temperature of 50 °C.'
- 'CPU #2 reports a temperature of 50 °C.'
- 'CPU #3 reports a temperature of 50 °C.'
- 'CPU #1 is consuming 100 W.'
- 'CPU #2 is consuming 100 W.'
- 'CPU #3 is consuming 110 W.'
- 'CPU #3 is consuming 90 W.'

Note that the last two sentences represent hypotheses, not data. The data consists in the fact that one of those two sentences is actually true.

The decision problem may require more basic sentences than just these, however. For instance, it might become necessary to consider sentences with other values for the temperature and of the power consumption, such as

'CPU #1 reports a temperature of 45 °C.'

The phrase "All three CPUs…" suggests that the basic sentence 'There are three CPUs' might be part of the data as well. Finally, there are obvious facts that we don't even think about, and yet it can become necessary to spell them out

---

[2]A more technical term is "atomic"

explicitly in some circumstances. An example here is the
sentence

'CPU #3 cannot be consuming both 110 W and 110 W.'

@@

We start by writing down the *basic*[3] sentences that constitute
our data and that underlie the inferences we want to draw.
"Basic" in the sense that we will not analyse these sentences
into further sub-sentences. In the trivial example above we
identify two such sentences: *"My umbrella is blue"*, and *"My
umbrella is pink"*. Let's represent them by symbols:

$$b := \text{"My umbrella is blue"}$$
$$r := \text{"My umbrella is red"}$$

Note a subtlety in our data – and again why we need to
make their underlying sentences as clear as possible: it
is understood here that my umbrella is all of one colour.

3.3.2 Connectives

You notice that we didn't consider *"My umbrella is either
blue or red"* and *"My umbrella is not red"* as basic sentences.
These sentences can indeed be expressed in terms of the ba-
sic sentences b and r. We consider one way or operation to
change a sentence into another related to it, and two ways or
operations to combine two or more sentences together. These
operations are called "connectives". Our natural language of-
fer many more operations to combine sentences, but these
three turn out to be all we need in virtually all engineering
problems. The three connectives are:

Not: $\neg$ for example,

$$\neg r = \text{"My umbrella is not red"}$$

And: $\wedge$ for example,

$$b \wedge r = \text{"My umbrella is blue, and it is red"}$$

---

[3]A more technical term is "atomic"

Or: ∨ for example,

$$b \vee r = \text{``My umbrella is blue, or red, or both''}$$

> Note some subtleties of the connectives:
>
> - "Not" doesn't mean some kind of complementary quality, but only the negation. For instance, ¬*"The chair is black"* does not mean *"The chair is white"*.
>
> - $b \vee r$ does not exclude, a priori, that my umbrella cannot be both blue and black (there is a connective for that: "exclusive-or", but it can be constructed out of the three we already have.)

From this last remark we see that the sentence *"My umbrella is either blue or red"* does not correspond to $b \vee r$. The sentence also means implicitly that my umbrella cannot be both blue or red. We could rewrite it as *"My umbrella is either blue or red, and it is not both blue and red"*. Convince yourself that in symbols we can write it like this:

$$(b \vee r) \wedge \neg(b \wedge r) = \text{``My umbrella is either blue or red''}$$

3.4 Combining assumptions and conclusions

"Basic" in the sense that we will not analyse these sentences into further sub-sentences. In the trivial example above we identify three such sentences: 'Rita has an umbrella', 'The umbrella is blue', and 'The umbrella is red'. Let's represent them by symbols:

$$b := \text{`Rita's umbrella is blue'}$$
$$r := \text{`Rita's umbrella is red'}$$

3.4.1 Connectives

We didn't consider 'Rita's umbrella is either blue or red', 'The umbrella is not blue', and 'The umbrella is not red' as basic sentences. These sentences can in fact be expressed in

terms of the basic sentences b and r. We consider one way or operation to change a sentence into another related to it, and two ways or operations to combine two or more sentences together. These operations are called **connectives**. Our natural language offers many more operations to combine sentences, but these three turn out to be all we need in virtually all engineering problems. The three connectives are:

Not: $\neg$ for example,

$$\neg b = \text{`Rita's umbrella is not blue'}$$

And: $\wedge$ for example,

$$b \wedge r = \text{`Rita's umbrella is blue, and it is red'}$$

Or: $\vee$ for example,

$$b \vee r = \text{`Rita's umbrella is blue, or red, or both'}$$

⚠ Note some subtleties of the connectives:

- "Not" doesn't mean some kind of complementary quality, but only the negation. For instance, $\neg$'Rita's umbrella is black' does not mean 'Rita's umbrella is white'.

- $b \vee r$ does not exclude, a priori, that the umbrella cannot be both blue and black (there is a connective for that: "exclusive-or", but it can be constructed out of the three we already have.)

- It is important to distinguish between *logical* impossibility and *physical* impossibility. It is physically impossible that an umbrella be fully white and fully black, but formal logic does allow us to consider the sentence "the umbrella is fully white and fully black", without setting it true or false a priori. We express its physical impossibility by assigning to it the value `false` in any inference we want to make. This generality of formal logic is a feature: it allows us to entertain and study hypotheses even if we still don't know whether they

> are physically impossible. Scientific research could not be done without this feature.

What about the "if" in "if the umbrella…"? Shouldn't that be a connective? This "if" is treated in a special way; we shall now see how it works.

Obviously we possess even more information, which is implicitly understood. For example, it's understood that the umbrella is all of one colour. It must also be understood what an umbrella is, what "red" and "blue" mean, and so on. Usually this kind of information is extremely obvious or irrelevant, so we don't think about it very much and don't mention it. In real engineering problems, however, it may happen that we have to stop and examine this implicit, hidden knowledge, maybe to check whether it contains contradictory information or contradictory goals. It's a good idea to represent this implicitly-understood information by a symbol, for example I (for "extra information"; we could have used K for "knowledge").

There is an old and still ongoing debate in logic and probability on how to deal with "if"-statements. Here we take the seemingly most common approach, which is simple and effective in applied problems.

3.4.2 Separating assumptions and conclusions

Now we have the sentences that represent all our data and the information we are interested in, and even symbols to represent them in a compact way. It's important to keep the assumptions we make well separated from the conclusions we want to draw. Let's introduce a symbol to do just that.

We can simply use a vertical bar: on its right side we write the sentences that make up our assumptions, and on its left side those that make up our desired conclusions.[4] So, in symbols, our inference is this:

$$b \,\big|\, (b \vee r) \,\wedge\, \neg r \,\wedge\, \neg(b \wedge r) \,\wedge\, D$$

Note how we `and`-ed all assumptions together. The collection of assumptions on the right side of the bar " $\big|$ " is called the **conditional**. The expression above is read "*b given* $(b \vee r)$" etc., or "*b conditional on* $(b \vee r)$" etc.

---

[4]Current notation in logic uses the symbols $\vDash$ or $\vdash$, and writes assumptions on the *left*, conclusions on the *right*. We use a different notation for an easier transition to probability logic.

All data in the conditional (that is, on the right side of "$\mid$") are considered (at least temporarily) to be `true`. Our goal is to determine whether the sentence on the left of "$\mid$" is `true` or `false`.

3.5 well-posed and ill-posed sentences

We face problems when the sentences that should convey information and data are not clear. Suppose that an electric-car model consumes 150 Wh/km and has a range of 200 km; a second car model consumes 250 Wh/km and has a range of 600 km. Someone says "I think the second model is better; what do you think?". It isn't clear how we should answer; what does "better" mean? If it refers to consumption, then the first car model is "better". If it refers to range, then the second model is "better". If it refers to a combination of these two characteristics, or to something else, then we simply can't answer. Here we have a problem with querying and giving data, because the sentence underlying such query is not clear.

We say that such sentences are **not well-posed**, or that they are **ill-posed**.

This may seem an obvious discussion to you. Yet you'd be surprised by how often unclear sentences appear in scientific papers about data engineering! Not seldom we find discussions and disagreements that actually come from unclear underlying sentences, that two parties interpret in different ways.

As a data engineer, you'll often have the upper hand if you are on the lookout for ill-posed sentences. Whenever you face an important question, or you're given an important piece of information, or you must provide an important piece of information, *always take a little time to examine whether the question or information is actually well-posed.*

@@ [TODO] Exercise: give actual paper to analyse_

Reading list

21

4 Inference

4.1 What is inference?

The first core problem in all data-driven engineering applications – and in daily life too – is to *draw inferences*, that is, acquire information. We may wish to acquire information out of simple curiosity, or for some specific engineering reason or goal, as we'll discuss later. Examples:

1. We'd like to know whether it'll rain today, so we can decide whether to get an umbrella or rain clothes.

2. A clinician would like to know which disease affects a patient, so as to decide for the optimal treatment.

3. The X-player of this game of Xs & Os: needs to know where put the next $\mathbf{X}$ in order to win.

4. The computer of a self-driving car needs to know whether a particular patch of colours in the visual field is a person, so as to slow down the car and stop.

5. In order to launch a rocket to the Moon, a rocket engineer needs to know, within two significant digits, how much is the velocity $\sqrt{2\,G\,M/r}$, where $G = 6.67 \cdot 10^{-11}\,\mathrm{m}^3\,\mathrm{s}^{-2}\,\mathrm{kg}^{-1}$, and $M = 5.97 \cdot 10^{24}\,\mathrm{kg}$ and $r = 6.37 \cdot 10^6\,\mathrm{m}$ are the mass and radius of the Earth.

6. We'd like to know whether the rolled die will show ⚄, so we can win a bet.

7. An aircraft's autopilot system needs to predict how much the aircraft's roll will change by increasing the right wing's angle of attack by 0.1 rad.

8. An archaeologist would like to know whether the fossil bone just dug out belonged to a Tyrannosaurus rex.

9. An automated system in an assembly line needs to predict whether an electric component of a widget will fail within the next two years.

Note how each of these inferences boils down to determining whether some sentences are true or false. In example 1. we want to know whether the sentence 'It rains today' is true or not. In example 2. the clinician wants to know which of the sentences 'The patient has pneumonia', 'The patient has asthma', 'The patient has bronchitis', and so on, are true (several can be true at the same time). In example 5. the rocket engineer wants to know which among the sentences 'The velocity is 0.010 m/s', 'The velocity is 0.011 m/s', …, 'The velocity is 130 m/s', and so on, is true. The sentences that underlie an inference can be extremely many and complex, and yet we must have an idea of what they are (otherwise, do we really know what our inference is about?).

> **Exercise**
>
> Try to identify which sentences underlie the other example inferences above.

4.2 Certain and uncertain inference

The example inferences above present very different levels of difficulty.

Inferences 3. and 5. are special because they can actually be drawn *exactly*, that is, we really find out which of their underlying sentences are true and false. In example 3. it is trivial that putting the next **X** in the mid-right slot makes the X-player win. In example 5. a couple of mathematical operations show that the sentence 'The velocity is 11 km/s' is true. When we can obtain the data we want from the data we have by using "only"[1] logic and mathematical operations, our inference is *certain*, also called a "deduction"; in these notes we shall call it a *truth inference*. But every deduction can be basically drawn by repeatedly applying the rules of logic.

---

[1] "Only" in quotation marks because the logical analysis and operations leading to the answer can still be computationally very expensive.

The other example inferences cannot be drawn exactly, in the sense that we cannot know for sure whether all their underlying sentences are true or false. But this doesn't mean that we cannot say anything whatsoever. In example 6. we consider the sentence 'The die shows six pips' to be more likely false than true. In example 2. the clinician might be quite sure about the disease, after observing the symptoms. On the other hand, in example 1. we might really have no clue whether 'It rains today' will turn out to be true or false. These inferences are *uncertain*. Certain inferences can be considered as a limit case of uncertain ones, in which the uncertainty vanishes or is extremely small.

To draw certain inferences, we follow the rules of Logic. What rules do we follow to draw uncertain inferences?

5 Truth inference

5.1 Building blocks

Consider the following trivial problem. An inspector examines an electronic component out of a production line. The information available to the inspector is the following:

- 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.

5.1.1 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'

5.1.2 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:

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

-

5.2 Background information and conditional

5.3 Truth-inference rules

Deduction systems in formal logic give us a set of rules for making correct inferences, that is, for correctly determining whether the conclusions of interest are true or false. These rules are represented in a wide variety of ways, as steps leading from one conclusion to another one. The picture here on the margin, for instance, shows how a proof of our inference would look like, using the so-called sequent calculus.

$$\frac{\dfrac{\dfrac{D \wedge \neg b \ \vdash \ \neg r}{D \ \vdash \ b \vee \neg r}}{\dfrac{D \ \vdash \ b \vee r \qquad D \wedge r \ \vdash \ b}{D \wedge D \ \vdash \ b \vee b}}}{\dfrac{D \wedge D \ \vdash \ b}{D \ \vdash \ b}}$$

Figure 5.1: The bottom formula is our conclusion; the formulae above it represent steps in the proof. Each line denotes the application of an inference rule. The two formulae with no line above are our two assumptions.

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:

$$\mathrm{T}(C \mid A) = 1 \ .$$

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

The rules of truth inference are then encoded by the following equations, which must always hold for any sentences A, B, C, no matter whether they are basic or complex:

Rule for "not":
$$\mathrm{T}(\neg A \mid B) + \mathrm{T}(A \mid B) = 1 \tag{5.1}$$

Rule for "and":
$$\mathrm{T}(A \wedge B \mid C) = \mathrm{T}(A \mid B \wedge C) \cdot \mathrm{T}(B \mid C) = \mathrm{T}(B \mid A \wedge C) \cdot \mathrm{T}(A \mid C) \tag{5.2}$$

Rule for "or":
$$\mathrm{T}(A \vee B \mid C) = \mathrm{T}(A \mid C) + \mathrm{T}(B \mid C) - \mathrm{T}(A \wedge B \mid C) \tag{5.3}$$

Rule of self-consistency:
$$\mathrm{T}(A \mid A \wedge C) = 1 \tag{5.4}$$

Let's see how the inference rule (**?@eq-example-rule**), for example, is encoded in these equations. The rule starts with saying that $a \wedge b$ is `true` according to D. This means that $\mathrm{T}(a \wedge b \mid D) = 1$. But, by rule (5.2), we must then have $\mathrm{T}(b \mid a \wedge D) \cdot \mathrm{T}(a \mid D) = 1$. This can only happen if both $\mathrm{T}(b \mid a \wedge D)$ and $\mathrm{T}(a \mid D)$ are equal to 1. So we can conclude that $\mathrm{T}(a \mid D) = 1$, which is exactly the conclusion under the line in rule (**?@eq-example-rule**).

> ✏ Exercise
>
> Try to prove our initial inference
>
> $$\frac{(b \vee r) \wedge \neg(b \wedge r) \mid D \qquad \neg r \mid D}{b \mid D}$$
>
> using the basic rules (5.1, 5.2, 5.3, 5.4). Remember that you can use each rule as many times as you like, and that there is not only one way of constructing a proof.

5.4 Logical AI agents and their limitations

The basic rules above are also the rules that a logical artificial-intelligent agent should follow.

> 📖 Reading
>
> Ch. 7 in *Artificial Intelligence*

Many – if not most – inference problems that a data engineer must face are, however, of the *uncertain* kind: it is not possible to surely infer the truth of some data, and the truth of some initial data may not be known either. In the next chapter we shall see how to generalize the logic rules to uncertain situations.

> 🚀 For the extra curious
>
> Our cursory visit of formal logic only showed a microscopic part of this vast field. The study of logic rules

continues still today, with many exciting developments and applications. Feel free take a look at *Logic in Computer Science*, *Mathematical Logic for Computer Science*, Natural Deduction Systems in Logic

6 Probability inference

6.1 When truth isn't known: probability

In most real-life and engineering situations we don't know the truth or falsity of sentences and hypotheses that interest us. But this doesn't mean that nothing can be said or done in such situations.

When we cross a busy city street we look left and right to check whether any cars are approaching. We typically don't look up to check whether something is falling from the sky. Yet, couldn't it be `false` that cars are approaching? and couldn't it be `true` that some object is falling from the sky? Of course both events are possible. Then why do we look left and right, but not up?

The main reason[1] is that we *believe strongly* that cars might be approaching, *believe very weakly* that some object might be falling from the sky. In other words, we consider the first occurrence to be very *probable*; the second, extremely improbable.

We shall take the notion of **probability** as intuitively understood (just as we did with the notion of truth). Terms equivalent for "probability" are *degree of belief*, *plausibility*, *credibility*.

> ❶ In technical discourse, *likelihood* means something different and is *not* a synonym of "probability", as we'll explain later.

Probabilities are quantified between `0` and `1`, or equivalently between `0%` and `100%`. Assigning to a sentence a probability `1` is the same as saying that it is `true`; and a probability

---

[1] We shall see later that one more factor enters the explanation.

`0`, that it is `false`. A probability of `0.5` represents a belief completely symmetric with respect to truth and falsity.

It is important to emphasize and agree on some facts about probabilities:

- **Probabilities are assigned to *sentences*.** Consider an engineer working on a problem of electric-power distribution in a specific geographical region. At a given moment the engineer may believe with `75%` probability that the measured average power output in the next hour will be 100 MW. The `75%` probability is assigned not to the quantity "100 MW", but to the *sentence*

  'The measured average power output in the next hour will be 100 MW'

  This difference is extremely important. Consider the alternative sentence

  'The average power output in the next hour will be *set* to 100 MW'

  the quantity is the same, but the meaning is very different. The probability can therefore be very different (if the engineer is the person deciding the output, the probability is `100%`). The probability depends not only on a number, but on what it's being done with that number – measuring, setting, third-party reporting, and so on. Often we still write simply '$O = 100\,\text{W}$' or even just '$100\,\text{W}$', provided that the full sentence behind the shorthand is understood.

- **Probabilities are agent- and context-dependent.** A coin is tossed, comes down heads, and is quickly hidden from view. Alice sees that it landed heads-up. Bob instead doesn't manage to see the outcome and has no clue. Alice considers the sentence 'Coin came down heads' to be `true`, that is, to have `100%` probability. Bob considers the same sentence to have `50%` probability.

  Note how Alice and Bob assign two different probabilities to the same sentence; yet both assignments are completely rational. If Bob assigned `100%` to 'heads', we would suspect that he had seen the outcome after all; if he assigned `0%` to 'heads', we would consider that groundless and silly. We would be baffled if Alice assigned `50%` to 'heads', because she saw the outcome was

actually heads; we would hypothesize that she feels unsure about what she saw.

An omniscient agent would know the truth or falsity of every sentence, and assign only probabilities 0 or 1. Some authors speak of "*actual* (but unknown) probabilities"; if there were "actual" probabilities, they would be all 0 or 1, and it would be pointless to speak about probabilities at all – every inference would be a truth inference.

- **Probabilities are not frequencies**. The fraction of defective mechanical components to total components produced per year in some factory is a quantity that can be physically measured and would be agreed upon by every agent. It is a *frequency*, not a degree of belief or probability. It is important to understand the difference between them, to avoid making sub-optimal decisions; we shall say more about this difference later. Frequencies can be unknown to some agents, probabilities cannot be unknown (but can be difficult to calculate). Be careful when you read authors speaking of an "unknown probability"; either they actually mean "unknown frequency", or a probability that has to be calculated (it's "unknown" in the same sense that the value of $1 - 0.7 \cdot 0.2/(1 - 0.3)$ is unknown to you right now).

- **Probabilities are not physical properties**. Whether a tossed coin lands heads up or tails up is fully determined by the initial conditions (position, orientation, momentum, rotational momentum) of the toss and the boundary conditions (air velocity and pressure) during the flight. The same is true for all macroscopic engineering phenomena (even quantum phenomena have never been proved to be non-deterministic, and there are deterministic and experimentally consistent mathematical representations of quantum theory). So we cannot measure a probability using some physical apparatus; and the mechanisms underlying any engineering problem boil down to physical laws, not to probabilities.

These facts are not just a matter of principle. They have important practical consequences. A data engineer who is not attentive to the source of the data (measured? set? reported, and so maybe less trustworthy?), or who does not carefully assess the context of a probability, or who mixes it up with something else, or who does not take advantage (when possible) of the physics involved in the engineering problem, will design a system with sub-optimal performance[2] – or even cause deaths.

6.2 No new building blocks

In discussing truth-inference we introduced notations such as $\mathrm{T}(a \mid b \wedge D)$, which stands for the truth-value 0 or 1 of sentence a in the context of data D and supposing (even if only hypothetically) sentence b to be true. We can simply extend this notation to probability-values, using a P instead of T:

$$\mathrm{P}(a \mid b \wedge D) \in [0, 1]$$

represents the probability or degree of belief in sentence a in the context of data D and supposing also sentence b to be true. Keep in mind that both a and b could be complex sentences (for instance $a = (\neg c \vee d) \wedge e$). Note that truth-values are included as the special cases1 or 0:

$$\mathrm{P}(a \mid b \wedge D) = 0 \text{ or } 1 \quad \Longleftrightarrow \quad \mathrm{T}(a \mid b \wedge D) = 0 \text{ or } 1$$

6.3 Probability-inference rules

Extending our truth-inference notation to probability-inference notation has been straightforward. But how do we draw inferences when probabilities are involved?

Consider the inference about my umbrella in a more uncertain situation:

---

[2]This fact can be mathematically proven.

$$\frac{\text{P(`My umbrella is either blue or red'} \mid D) = 1 \quad \text{P(`My umbrella is not red'} \mid D) = 0.5}{\text{P(`My umbrella is blue'} \mid D) = ?}$$

or more compactly, using the symbols we introduced earlier,

$$\frac{\mathrm{P}\big[(b \vee r) \wedge \neg(b \wedge r) \mid D\big] = 1 \quad \mathrm{P}(\neg r \mid D) = 0.5}{\mathrm{P}(b \mid D) = ?}$$

This says, above the line, that: according to our data D my umbrella is either blue or red (and can't be both), with full certainty; and according to our data we have no preferential beliefs on whether my umbrella is not red. What should then be the probability of my umbrella being blue, according to our data?

Intuitively that probability should be 50%: $\mathrm{P}(b \mid D) = 0.5$. But which rules did we follow in arriving at this probability? More generally, which rules should we follow in assigning new probabilities from given ones?

The amazing result is that *the rules for truth-inference, formulae (5.1, 5.3, 5.2, 5.4), extend also to probability-inference.* The only difference is that they now hold for all values in the range $[0, 1]$, rather than only values 0 and 1.

This important result was taken more or less for granted at least since Laplace in the 1700s. But was formally proven for the first time in the 1940s by R. T. Cox; the proof has been refined since then. What kind of proof is it? It shows that if we don't follow the rules we arrive at illogical conclusions; we'll show some examples later.

Here are the fundamental rules of probability inference. In these rules, all probabilities can have values in the range $\mathrm{P}() \in [0, 1]$, and the symbols a, b, D represent sentences of any complexity:

It is amazing that **ALL** inference is nothing else but a repeated application of these four rules – billions of times or more, in some inferences. All machine-learning algorithms are just applications or approximations of these rules. Methods that you may have heard about in statistics are just specific applications of these rules. Truth inferences are also special applications of these rules. Most of this course is, at

> **🏛 THE FUNDAMENTAL LAWS OF INFERENCE 🏛**
>
> **"Not" ¬ rule**
> $$P(\neg a \mid D) + P(a \mid D) = 1$$
>
> **"And" ∧ rule**
> $$P(a \wedge b \mid D) = P(a \mid b \wedge D) \cdot P(b \mid D) = P(b \mid a \wedge D) \cdot P(a \mid D)$$
>
> **"Or" ∨ rule**
> $$P(a \vee b \mid D) = P(a \mid D) + P(b \mid D) - P(a \wedge b \mid D)$$
>
> **Self-consistency rule**
> $$P(a \mid a \wedge D) = 1$$

bottom, just a study of how to apply these rules in particular kinds of problems.

> **📖 Reading**
>
> - *Probability, Frequency and Reasonable Expectation*
>
> - Ch. 2 of *Bayesian Logical Data Analysis for the Physical Sciences*
>
> - §§ 1.0–1.2 of *Data Analysis*
>
> - Feel free to skim through §§ 2.0–2.4 of *Probability Theory*

6.4 How the inference rules are used

The fundamental rules represent, first of all, constraints of logical consistency among probabilities. If we have probabili-

ties $P(a \mid D) = 0.7$, $P(b \mid a \wedge D) = 0.1$, $P(a \wedge b \mid D) = 0.2$, then there's an inconsistency somewhere, because these values violate the and-rule: $0.2 \neq 0.1 \cdot 0.7$. In this case we must find the inconsistency and solve it. Since probabilities are quantified by real numbers, however, it's possible and acceptable to have slight discrepancies owing to numerical round-off errors.

The rules also imply more general constraints. For example we must *always* have

$$P(a \wedge b \mid D) \leq \min\{P(a \mid D), \ P(b \mid D)\}$$
$$P(a \vee b \mid D) \geq \max\{P(a \mid D), \ P(b \mid D)\}$$

> ✏️ Exercise
>
> Try to prove the two constraints above

The main use of the rules in concrete applications is for calculating new probabilities from given ones. The calculated probabilities will be automatically consistent. For each equation shown in the rules we can calculate one probability given the remaining ones in the equation, with some special cases when values of 0 or 1 appear.

For example, if we have $P(a \wedge b \mid D) = 0.2$ and $P(a \mid D) = 0.7$, from the and-rule we can find $P(b \mid a \wedge D)$:

$$\underbrace{P(a \wedge b \mid D)}_{0.2} = P(b \mid a \wedge D) \cdot \underbrace{P(a \mid D)}_{0.7}$$

$$\implies \quad P(b \mid a \wedge D) = \frac{P(a \wedge b \mid D)}{P(a \mid D)} = \frac{0.2}{0.7} \approx 0.2857$$
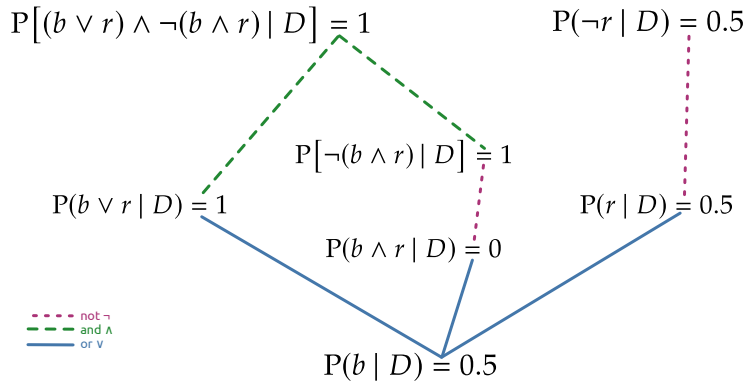
Let us now solve the umbrella inference from the previous section. Starting from

$$P\big[(b \vee r) \wedge \neg(b \wedge r) \mid D\big] = 1 \ , \quad P(\neg r \mid D) = 0.5$$

we arrive at

$$P(b \mid D) = 0.5$$

by following from top to bottom the steps depicted here:

$\mathrm{P}\big[(b \vee r) \wedge \neg(b \wedge r) \mid D\big] = 1$

$\mathrm{P}(\neg r \mid D) = 0.5$

$\mathrm{P}\big[\neg(b \wedge r) \mid D\big] = 1$

$\mathrm{P}(b \vee r \mid D) = 1$

$\mathrm{P}(r \mid D) = 0.5$

$\mathrm{P}(b \wedge r \mid D) = 0$

not ¬
and ∧
or ∨

$\mathrm{P}(b \mid D) = 0.5$

@@ example medical diagnosis

6.4.1 Derived rules

The rules above are in principle all we need to use. But from them it is possible to derive some additional shortcut rules that are automatically consistent with the fundamental ones.

First, it is possible to show that all rules you may know from Boolean algebra are a consequence of the fundamental rules. For example, we can always make the following convenient replacements anywhere in a probability expression:

$$A \wedge A = A \vee A = A \qquad \neg\neg A = A$$

$$A \wedge B = B \wedge A \qquad A \vee B = B \vee A$$

$$\neg(A \wedge B) = \neg A \vee \neg B \qquad \neg(A \vee B) = \neg A \wedge \neg B$$

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

Two other derived rules are used extremely often, so we treat them separately.

6.5 Law of total probability or "extension of the conversation"

6.6 Bayes's theorem

6.7 consequences of not following the rules,

@@ §12.2.3 of AI

- *Exercise: Monty-Hall problem & variations*

- *Exercise: clinical test & diagnosis*

Figure 6.1: Bayes's theorem guest-starring in *The Big Bang Theory*

6.8 Common points of certain and uncertain inference

No premises? No conclusions!

❗ Differences in terminology

- Some texts speak of the probability of a "random[3] variable", or more precisely of the probability that a random variable takes on a particular value. As you notice, we have just expressed that idea by means of a *sentence.* The viewpoint and terminology of random variables is a special case of that of sentences. As already discussed, in concrete applications it is important to know how a variable "takes on" a value: for example it could be directly measured, indirectly reported, or purposely set. Thinking in terms of sentences, rather than of random variables, allows us to account for these important differences.

- Some texts speak of the probability of an "event". For all purposes an "event" is just what's expressed in a sentence.

It's a question for sociology of science why some people keep on using less flexible points of view or terminolo-
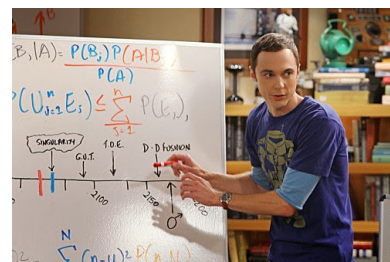
---

gies. Probably they just memorize them as students and then a fossilization process sets in.

---

[3]What does "random" mean? Good luck finding an understandable and non-circular definition in texts that use that word. In these notes, if the word "random" is ever used, it means "unpredictable" or "unsystematic".

7 Data and information

7.1 Kinds of data

7.1.1 Binary

7.1.2 Nominal

7.1.3 Ordinal

7.1.4 Continuous

- unbounded
- bounded
- censored

7.1.5 Complex data

2D, 3D, images, graphs, etc.

7.1.6 "Soft" data

- orders of magnitude
- physical bounds

7.2 Data transformations

- log
- probit
- logit

8 Allocation of uncertainty among possible data values: probability distributions

8.1 The difference between Statistics and Probability Theory

Statistics is the study of collective properties of collections of data. It does not imply that there is any uncertainty.

Probability theory is the quantification and propagation of uncertainty. It does not imply that we have collections of data.

8.2 What's "distributed"?

Difference between distribution of probability and distribution of (a collection of) data.

8.3 Distributions of probability

8.3.1 Representations

- Density function
- Histogram
- Scatter plot

Behaviour of representations under transformations of data.

8.4 Summaries of distributions of probability

8.4.1 Location

Median, mean

8.4.2 Dispersion or range

Quantiles & quartiles, interquartile range, median absolute deviation, standard deviation, half-range

8.4.3 Resolution

Differential entropy

8.4.4 Behaviour of summaries under transformations of data and errors in data

8.5 Outliers and out-of-population data

(Warnings against tail-cutting and similar nonsense-practices)

8.6 Marginal and conditional distributions of probability

8.7 Collecting and sampling data

8.7.1 "Representative" samples

Size of minimal representative sample $= (2\hat{}\text{entropy})/\text{precision}$

- *Exercise: data with 14 binary variates, 10000 samples*

8.7.2 Unavoidable sampling biases

In high dimensions, all datasets are outliers.

Data splits and cross-validation cannot correct sampling biases

8.8 Quirks and warnings about high-dimensional data

9 Making decisions

9.1 Decisions, possible situations, and consequences

9.2 Gains and losses: utilities

9.2.1 Factors that enter utility quantification

Utilities can rarely be assigned a priori.

9.3 Making decisions under uncertainty: maximization of expected utility

10 The most general inference problem