Lecture 2

Cezar Ionescu

Probability theory

- The first question of probability theory is "probability of what?".
  - Richard von Mises (1883-1953): "The term probability is meaningful for us only with regard to a clearly defined collective[1] (or population." (*Mathematical Theory of Probability and Statistics*, page 17).
  - Sir Harold Jeffreys (1891-1989): "there is a valid primitive idea expressing the degree of confidence that we may reasonably have in a proposition" (*Theory of Probability, 3rd Ed.*, page 15).
- The currently accepted mathematical model was developed by Kolmogorov in 1933 and covers both interpretations.
  - Ingredients:
    * `Ω`, the set of all possible outcomes
    * `Event`, the set of all events `Event ⊆ ℙ(Ω)`[2], which are what we actually want to assign probabilities to. The set of events is assumed to satisfy certain elementary properties (forming a σ-algebra of `Ω`).
    * `p : Event → [0, 1]`, the probability measure, satisfying
      1. `p(∅) = 0, p(Ω) = 1`
      2. for all events `X₁, ..., Xₙ, ...` pairwise disjoint, `Σ p(Xᵢ) = p(⋃ Xᵢ)`

Examples

1. Drawing a card from a standard deck. Any of the 52 cards is a possible result, so `Ω = {2♣, 2♦, 2♥, 2♠, ...}`. An event is what we want to assign a probability to, for example "the card drawn is a spade", or "a red card between 7 and 10". Any event can be represented by a subset of `Ω`; when `Ω` is finite, we take *all* subsets, so that `Event = ℙ(Ω)`.
2. Choosing a point of the unit disc. This can be seen as an idealised model for a game of darts. `Ω = {(x, y) | x, y ∈ ℝ, x² + y² ≤ 1}`. Events are of the form "the point will be chosen from this or that subset of the unit disc", so in principle we would

[1] Intuitively, a *collective* is an infinite sequence of results of a repeated experiment.
[2] `ℙ(X)` denotes the set of all subsets of `X`, including `∅` and `X` itself.

like again to have `Event = ℙ(Ω)`. It is an annoying but inescapable mathematical fact that we cannot, since that would make it impossible to define a probability measure satisfying the conditions above. Therefore, we limit the number of subsets that we assign probabilities to. The standard mathematical choice is that of a *Borel σ-algebra*, which is "almost as big" as `ℙ(Ω)`. We can be even less ambitious and choose `Event` to be the set of all *computable* subsets of `Ω` (i.e., all those whose characteristic function `X : Ω → {0, 1}` can be implemented as a program).

The classical model of probability

`Ω` is finite, `Event = ℙ(Ω)` and the probability measure is defined by

```
p(X) = card(X) / card(Ω)
```

where `card(X)` is the number of elements in the finite set `X`.

Interpretations:

- Frequentist: if we were to repeat the experiment many times, every elementary outcome `ω ∈ Ω` would turn up with approximately the same frequency as any other.
- Bayesian: in the absence of any information, we have no reason to prefer one elementary outcome over any other, so we must have `p(ωᵢ) = p(ωⱼ)` for all `i, j`.

Example: rolling a die

- `Ω = {1, 2, 3, 4, 5, 6}, card(Ω) = 6`
- rolling an even number: `even = {2, 4, 6}, card(even) = 3`
- probability of rolling an even number: `p(even) = card(even)/card(Ω) = 3/6 = 0.5`

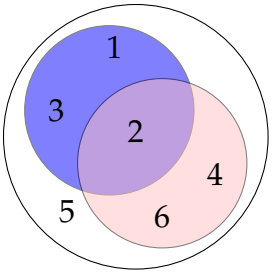
Operations with events:

Since events and predicates are modelled as sets, we can perform on them set-theoretical operations.

- union: `A ∪ B`
- intersection: `A ∩ B`
- complement: `¬ A = Ω - A`

Examples:

- Example: probability that the result is even **and** `≤ 3`
- Example: probability that the result is even **or** `≤ 3`
- Example: probability that the result is **not** `≤ 3`

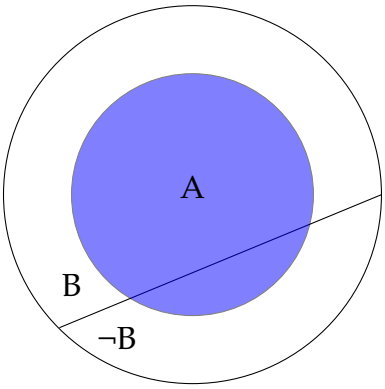


Remark: Operations on sets correspond to logical operations.

Exercise:

1. Derive the formula for `p(¬A)`.
2. Derive the formula for `p(A ∪ B)`
3. Assume that `B₁, ..., Bₙ` are pairwise disjoint. Show that

```
p(A) = p(A ∩ B₁) + ... + p(A ∩ Bₙ)
```



Conditional probability

The following definition is perhaps the most misunderstood aspect of elementary probability theory.

Definition: Let `(Ω, Event, p)` as above, and let `X ∈ Event` such that `p(X) ≠ 0`. Assume that event `X` has been realised (the elementary result of an experiment, `ω`, is an element of `X`). The probability that another event, `Y ∈ Event`, has *also* been realised is denoted by `p(Y | X)` and defined as

```
p(Y | X) = p(Y ∩ X) / p(X)
```

The notation `p(Y | X)` **is very bad**. It suggests that `|` is a set-theoretical operation, just like `∪` or `∩`. After all, `p` is defined on events, and it seems to be applied to `Y | X`, which therefore must be an event, hence a subset of `Ω`. The notation also suggests a symmetry between `X` and `Y` which does not exist: the two events have completely different roles. The notation used by Kolmogorov in his original publication of 1933 was $p_X(Y)$. This makes it clear that we are dealing with **another probability measure**, namely, one that reflects the

assumption that `X` has been realised, and there is no hint of symmetry. Unfortunately, the inferior notation has been universally adopted, and we have no choice but to follow suit.

Exercise (the law of total probability): Assume that `B₁, ..., Bₙ` are pairwise disjoint, such that `p(Bᵢ) ≠ 0` for all `i`. Show that

```
p(A) = p(A | B₁)*p(B₁) + ... + p(A | Bₙ)*p(Bₙ)
```

Bayesian learning

Recall the setting from lecture 1. We have a set of hypotheses, `H`, and are given some data, `d ∈ D`. We want to find the "best" hypothesis that fits the data. The question is: what does "best" mean? In the context of `Find-S`, "best" meant the most specific hypothesis consistent with the data. In the much wider (and more reasonable) context of Bayesian learning, "best" means *the most probable* hypothesis, given the data.

In other words, the aim of Bayesian learning is to determine the **maximum a posteriori (MAP) hypothesis**, i.e., the hypothesis that satisfies

```
h_map = argmax p(_ | d)
```

Remarks:

1. The partial function `argmax` satisfies

```
argmax : (X → ℝ) → X
argmax f = x iff for all x' ∈ X f(x) ≥ f(x')
```

2. The notation `p(h | d)` is an abbreviation for `p({h} | {d})` (remember that events are *sets*).

3. The expression `p(_ | d)` denotes a function of argument `h`:

```
p(_ | d) : H → [0, 1]
p(_ | d)(h) = p(h | d)
```

4. Alternative notations for `p(_ | d)` are `h ↦ p(h | d)`, so we could have written

```
h_map = argmax (h ↦ p(h | d))
```

   or

```
h_map = argmax (h ∈ H ↦ p(h | d))
```

   to make the domain of the function explicit. In the context of `argmax`, the most frequent notation is

```
h_map = argmax_h p(h | d)
```

Exercise:

- What specification does the partial function `max` satisfy?
- Why are `max` and `argmax` partial?

Bayes' theorem

Consider events `X`, `Y`, such that `p(X) * p(Y) ≠ 0`. From the definition of conditional probability, we have

```
p(X | Y) = p (X ∩ Y) / p (Y), therefore p(X ∩ Y) = p(X | Y) * p(Y)
p(Y | X) = p (Y ∩ X) / p (X), therefore p(Y ∩ X) = p(Y | X) * p(X)
```

But `X ∩ Y = Y ∩ X`, therefore `p(X ∩ Y) = p(Y ∩ X)`, therfore

```
p(X | Y) = p(Y | X) * p(X) / p(Y)
```

This relatively trivial result is sometimes called "Bayes' theorem".

Example: (Mitchell, pages 157-158)

- `H = {healthy, ill}`
- There is a test with possible results `⊖`, `⊕` ("negative", i.e., the disease is not present, and "positive", i.e., the disease is present).
- We are told that only `0.008` of the population have the disease.
- We are also told that the test is not perfectly reliable. If the patient has the disease, the test will be positive only in `98%` of the cases. If the patient does not have the disease, then the test will still be positive in `3%` of the cases.
- Finally, we are given the data: a patient's test has come back positive. The question is: what is the MAP hypothesis? Is it likelier that the patient is ill, or healthy?

We need express the problem in the language of probability theory. That is, we must find `Ω, Event, p`. This is, at least in simple situations, the most difficult (and important) step. For example:

- `Ω` = the set of people in the population
- since `Ω` is finite, `Event = ℙ(Ω)`
- the experiment consists of drawing a random person from the population, with equal probability, and applying the test. Therefore, the classical model applies: `p(X) = card(X)/card(Ω)`
- the events of interest are: `Healthy`, the subset of the population who are healthy, `Ill`, the subset of the population whose members have the disease, `Pos`, the subset of the population for which the test result was positive, `Neg`, for which the test was negative.
- the given percentages are translated as follows: `p(healthy) = 0.992`, `p(ill) = 0.008`, `p(⊕ | ill) = 0.98`, `p(⊕ | healthy) = 0.03`
- the question we are asked is to compute `p(ill | ⊕)`.

We can now try to answer the question by applying Bayes' theorem:

```
    p(ill | ⊕)
=   p(⊕ | ill) * p(ill) / p(⊕)
=   0.98 * 0.008 / p(⊕)
=   0.00784 / p(⊕)
```

```
    p(healthy | ⊕)
=   p(⊕ | healthy) * p(healthy) / p(⊕)
=   0.03 * 0.992 / p(⊕)
=   0.02976 / p(⊕)
```

Therefore, the MAP hypothesis is `healthy`.

In general, we have

```
h_map = argmax_h p(h | d)
      = argmax_h (p(d | h) * p(h) / p(d))
      = argmax_h (p(d | h) * p(h)) -- since p(d) is constant w.r.t. h
```

If, additionally, we assume that all hypotheses are equally likely, i.e., `p(h)` is also constant w.r.t. `h` (`p(h) = 1/card(H)` when `H` is finite), then we can go one step further and obtain

```
h_map = argmax_h p(h | d)
      = argmax_h p(d | h)
```

`p(d | h)`, the probability of encountering the data `d` if we assume that hypothesis `h` is correct, is called the **likelihood** of `d` given `h`. A hypothesis that maximises the likelihood is known, reasonably enough, as a *maximum likelihood hypothesis (ML)*. Therefore

```
h_ml = argmax_h p(d | h)
```

Please note that, in general, $h_{map} \neq h_{ml}$! The equality only holds when we assume that all hypotheses are equally likely! In particular, in the example above, this is not the case.

Exercise: what is h_{ml} in the example above?

Homework

Apply Bayes' theorem to answer the following question (Elmer Mode 1966, page 53):

> A class in advanced mathematics contains 10 juniors, 30 seniors, and 10 graduate students. Three of the juniors, 10 of the seniors, and 5 of the graduate students received an A in the course. If a student is chosen at random from this class and is found to have earned an A, what is the probability that he is a graduate student?

Bayesian concept learning

Recall that a concept is a subset of, or a boolean function defined on, a given set:

```
c : X → {0, 1}
```

Each hypothesis `h ∈ H` is such a subset or function (`h : X → {0, 1}`), and we assume that `c ∈ H`.

The data we are given in the concept learning task consists of pairs

```
d = {(x₁, c(x₁)), ..., (xₘ, c(xₘ))}
```

The "brute-force" Bayesian concept learning algorithms then returns a (or "the") MAP hypothesis:

```
given H, p(h)
for every h ∈ H:
  compute p(h | d) = p(d | h) * p(h) / p(d)
  return h_map = argmax_h p(h | d)
```

The problem is computing `p(d | h)` and `p(d)`. As we have seen above, the latter is not necessary, but it is very instructive.

To compute the likelihood, we make the following assumption: the data is completely correct, and hypotheses are not "noisy" functions. Therefore, for any `h ∈ H`, if we assume that `h = c`, then the data `d` can only arise if `h` is consistent with it. Therefore:

```
p(d | h) = if for all (xᵢ, bᵢ) ∈ d  h(xᵢ) = bᵢ
              then 1
              else 0
```

Now that we have computed `p(d | h)` for every `h ∈ H`, we can compute the a-priori probability that the data `d` is realised by applying the law of total probability:

```
p(d) = Σ_h p(d | h) * p(h)
```

This computation shows that the a-priori estimate of the probability of data must be consistent with the a-priori estimate of the probability of hypotheses. Normally, we consider data to be more *objective* than the hypotheses, which are more *subjective*, so this can cause some confusion.

Bayesian concept learning and `Find-S`

If we assume that all hypotheses are equally likely, then the brute-force Bayesian concept learning algorithm will return an ML hypothesis:

```
h_map = h_ml = argmax_h p(d | h)
```

Since `p(d | h) = 1` if `h` is consistent with the data, and `p(d | h) = 0` otherwise, the algorithm will return any one of the hypotheses in `H` consistent with `d`.

Since `Find-S` returns a consistent hypothesis, it therefore follows that it returns a MAP (and an ML) hypothesis too!

Even if `p` is not constant w.r.t. hypotheses, `Find-S` can still return a MAP hypothesis, if

```
hᵢ ⊆ hⱼ ⇒ p(hᵢ) ≥ p(hⱼ)
```

(i.e., `p` favours the more specific hypothesis), and if there is a unique most specific hypothesis consistent with `d` in `H`.