EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Supplementary lab – intro to probability

Or: Combinatorics on steroids

- *The Language of Thought: computational cognitive science approaches to category learning*
- Who: Fausto Carcassi
- When: Sommer semester 2022

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Probability – finite discrete

- Suppose we flip a coin three times.
  - What is the probability of getting tails exactly once?
- Suppose we flip a coin 30 times.
  - What is the probability of getting tails exactly 5 times?
- Suppose we roll a dice twice.
  - What is the probability of the sum of the two values being 4?
- Note the ingredients of these questions:
  - We have **some situation** with a probabilistic component
  - The situation can turn out in various **ways**
  - We are interested in the **probability** of each of those ways

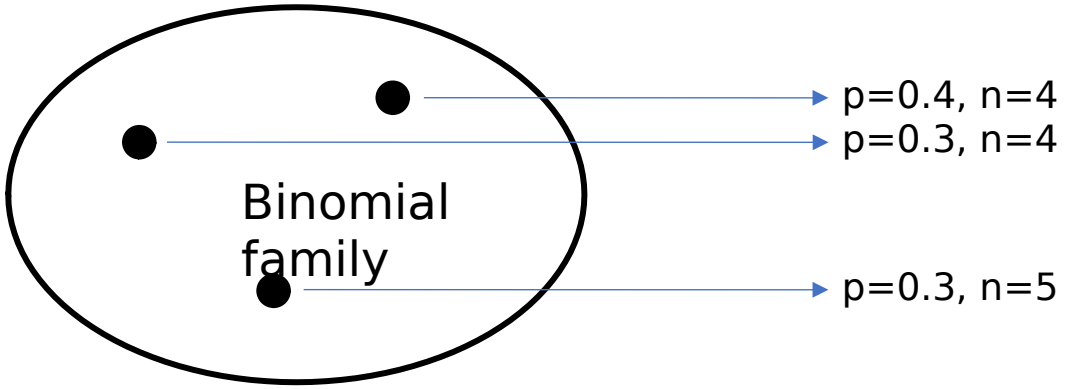
Probability – finite discrete

Often we need one of four distributions in this context:

- Binary event {0,1} that happens once: **Bernoulli distribution**
  - Only has one parameter *p*, namely the probability of 1
- Binary event {0,1} that happens *n* times: **Binomial distribution**
  - Answers the question: What is the p that 0 happens exactly n times?
  - Has parameters *p* and *n*.
  - Bounded above by *n*: [0, *n*] is called the *support* of the distribution
- Event {0,1,…,*m*} that happens once: **Categorical distribution**
  - Has one parameter, which is *not a number!*, but a prob vector of length *m*
  - Question: why is *m* not a parameter?
  - Can you tell me what a probability vector is?
  - What question do you think it answers?
- Event {0,1,…,*m*} that happens *n* times: **Multinomial distribution**
  - Two parameters, *n* and a probability vector of length *m*
  - What question do you think it answers?

Probability – families

- We have been talking vaguely, so let’s introduce some terminology
- The various ‘distributions’ we have talked about are not distributions. They become distributions when a value is given for the parameters.
  - Rather, they are called *families* of distribution.
  - For instance, ‘Bernoulli family’, ‘Binomial family’, etc.
  - You can think of a family of distributions as a *parameterized set*: a set of distributions, each of which can be specified by setting some parameters.
  - Infinite sets!



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Probability – support

- The *support* of a distribution is the set of values with probability > 0
- In many cases, the support is a set of numbers
  - For instance, in Bernoulli distributions the support is {0, 1}
- However, the support doesn't have to be a set of numbers
  - Can you think of an example where it's not a set of numbers?
  - For instance, the support of the multinomial distribution is the set of non-negative integer vectors summing to n: Vectors, not numbers!
- Question: does the support have to be identical for all distributions in a given family?
  - Answer: No!
  - E.g. for binomial and multinomial, the support depends on n

Probability mass function

- I have talked about the *distributions* as giving us probabilities of events, but this is not quite right.
  - A probability distribution is an abstract object with various properties, like a *support* etc.
  - Think about it as an object in python.
  - What gives us the prob of an event is one of the things associated with the distribution: its *probability mass function*.
- We might want to get other information about the distribution.
  - For instance, ‘what is the probability of all elements in the support that are greater than 3?’ for a binomial distribution with n3.
  - This is called the *cumulative distribution function*
  - Distributions define loads of other functions too!

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Random variables

- *Random variables* are a way of making all of this more formal.
- They're usually not taught in intro courses because they're a bit technical.
- However, not understanding them leads to terrible confusions
- A random variable is a *function* (so it's neither a variable nor random!)
  - The support of this function is the *sample space* : a set of *events* with associated probabilities.
  - The range of the function is a set of number or vectors
  - (Range can be other things too but we'll only need those two)
  - (I'm skipping *a lot* of technical detail here)
- Random variables model quantities that have a distribution, like:
  - 'the number of heads of a coin flipped twice'
  - 'the number of times a random person has seen the moon'

Random variable - examples

- ‘whether a flipped coin shows heads’
- ‘whether a flipped coin shows heads of tails’
- ‘the number of heads of two flipped coins’
- ‘the number of leaves in a tree in bota’
- ‘whether the number of leaves in a tree in bota is odd’

Random variables

- We say that a random variable *has a certain distribution* or that *it is distributed as a certain distribution* and we write:
  - Random variable its distribution
- Note that we are using *the distribution*, the abstract object
- For instance,
  - Call X the total number of heads from flipping a fair coin four times
  - Then we can write:
  - And we say 'X is distributed as a Binomial with n parameter 4 and p parameter 0.5'

Probability – continuous support

- In order to explain the basic terminology, I have only used distributions with finite discrete support.
- But of course, there are distributions with
  - Infinite discrete support, e.g., [0, 1, 2, …,)
  - Bounded continuous support, e.g., [0, 1]
  - Unbounded continuous, e.g., [0,)
- It’s a bit harder to deal with them mathematically, but what is important now is that you understand what’s going on conceptually.
- Do you know examples of these?

If there is time left: some foundations

- What is a set?
  - Abstract collection of disjoint (non-repeated) objects
- How can we define a set?
  - Listing the elements of the set (but careful: sets are not ordered!)
    - List notation: {table, chair}
  - Stating a property which all and only the objects in the set have
    - Intentional notation: {x | x is an integer greater than 3}
  - Defining rules that generate the elements of the set
    - Recursive definition you have seen in the python lab!
- Two sets are the same iff they contain the same members
- Member, subset, superset, intersection, union, power set

If there is time left: some foundations

- What is an ordered tuple?
  - <a, b> = {{a}, {a, b}}
  - <a, b, c> = <<a, b>, c>
- Cartesian product of two sets A and B

- What is a relation?
  - A *relation from set A to set B* is any subset of
- What is a function?
  - A *function from A to B* is a relation R from A to B such that:
  - Each element of A appears exactly once in the first elements of R.