A probabilistic semantics for natural language

Jean-Philippe Bernardy

First Tbilisi International Summer School in "Logic, Language, Artificial Intelligence"

Contents

1 Motivation

- All men are mortal
- Socrates is a man
- Socrates is mortal?

vs.

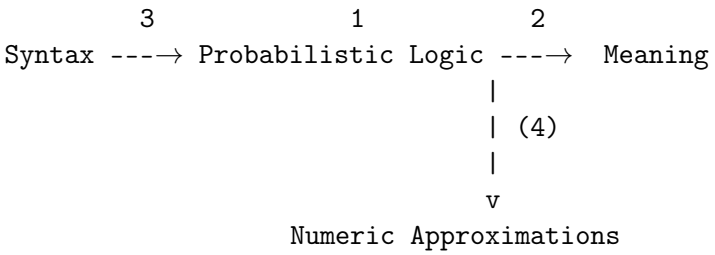
- John is taller than many basketball players
- Basketball players are generally tall
- John is tall?

1.1 For the Cognoscenti

- Probabilistic reasoning is thought to occur in everyday life and events.
- It has proven useful to model various linguistic phenomena (graded adjectives, pragmatics, etc.)
- Classical Bayesian reasoning and vector models can be combined. Deep learning models have shown that individuals/situations and even predicates can be represented as points in a large-dimensional euclidean space (e.g. cosine distance). Hypothesis: Bayesian models can model such spaces.
- Intuitive probabilistic syllogisms can be accurately modeled.

2 Plan

Preamble: elements of probability theory.



3 Elements of Probability theory

3.1 Introductory Problem

Assume a big bag, which you know contains a lot of red and blue balls. The contents of the bag is thoroughly mixed. You do not know the proportion of blue and red balls in the bag.

You pick 4 balls at random, putting each ball back in the bag after looking at it. The first three are red, the last one is blue.

What is the probability for your next ball pick to yield a red ball?

3.2 Concept: Probability distribution

1. Frequency distribution Consider a coin, with two faces, nominally labelled "heads" and "tails".

   Throw it n times. Consider what you will get.

   - f(Heads) + f(Tails) = n

   Consider a die, with 6 faces. Throw it n times. Consider what you will get.

   Let $\Omega = \{1,2,3,4,5,6\}$

   Note:

   - $\sum(x : \Omega)f(x) = n$

   We say that Ω is the probability space of x.

2. Discrete probability distributions

   One can define the probability of an event P(x) (with x:Ω) as the limit of a frequency distribution f divided by the total number of observations of x, for the number of observations tending to infinity.

   If the coin is "fair", we then expect:

   - P(Heads) = 0.5
   - P(Tails) = 0.5

   For any probability distribution P, given set of disjoint events Ω, we expect:

   - $\sum(x : \Omega)P(x) = 1$

3. Axiomatic view of probability distribution

   Instead we will take the axiomatic view and say that any function P summing to 1 over a set Ω is a probability distribution.

   Note that Ω needs not be finite (but in this case lots of P(x) will be infinitesimal).

4. Continuous probability distribution Later on, we will mostly turn our attention to set of events Ω which are not discrete. For example, instead of considering whether the coin falls heads or tails, consider *where* it ends up falling, for example as a pair of coordinates.

   - $\Omega = \mathbb{R}^2$

   If we'd attempt to use a probability distribution as before, we end up with P(x) = 0 for every point. So in this case, each element of Ω is assigned not a probability but a *probability density*.

5. Continuous probability distribution: properties

   If f is the *probability density function* (PDF) of P, the fundamental property becomes:

   - $\int(x : \Omega)f(x)dx = 1$

   Remark: we'll almost never care about the value of f directly; only its behaviour under integrals. That is, the only valid question to ask is the probability of the coin falling "within an area" — not "exactly" at a given point.

3.3 Notation

We can generalise the notation P(...) as follows:

- If C is a subset of Ω, then
  - $P(C) = \sum(x : C)P(x)$ if Ω is discrete
  - $P(C) = \int(x : C)PDF(x)$ dx if Ω is continuous
- If c is a Boolean expression (dependening on x),
  - $P(c) = P(x \in \Omega|c)$.

  That is, we check the probability of the set of events which makes c true.

1. Examples
   - P(Heads $\cup$ Tails)
   - P({d $>$ 3 | d $\in$ {1,2,3,4,5,6}})

3.4 Dependent and Independent events and variables

- We say that events A and B are independent iff.
  - $P(A \wedge B) = P(A) \cdot P(B)$.
- The probability P(A $\cap$ B) is **not** equal to P(A) $\cdot$ P(B) in general!

3.5 Examples

1. Coins
   - $P(Heads \wedge Tails) = 0$ (Indeed, the events are *dependent* on each other)
2. Dice
   - Throw a pair of 6-faced dice d1, d2. P(d1+d2 $>$ 9) = ?

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

- 36 (equally probable) cases
- 6 satisfy the condition
- $\rightarrow$ P(d1+d2 $>$ 9) = 6/36 = 1/6

3.6 Conditional probability (1)

Definition:

- $P(A|B) = P(A \wedge B)/P(B)$
  - if $P(B) > 0$

Example:

```
P (Heads | Tails) = P (Heads ∧ Tails) / P(Head)
                  = 0 / 0.5
                  = 0

P (d1+d2 > 9 | d2=5) = P(d1+d2 > 9 ∧ d2=5) / P(d2 = 5)
                     =  (2 / 36)           / (1/6)
                     =   1 / 3
```

3.7 Conditional probability (2)

Alternatively one can use P(A | B) as a primitive notion and define

- $P(A \wedge B) = P(A|B) \cdot P(B)$

This equation is useful when P(A | B) is known or easy to compute.

1. Example

```
       P(d1+d2 > 9 ∧ d2=5)
-- by the above
     = P (d1+d2 > 9 | d2=5) · P(d2=5)
-- by substitution
     = P (d1+5 > 9)  · P(d2=5)
-- by subtracting 5
     = P (d1 > 4)  · P(d2=5)
     = (2/6)       · (1/6)
     = (2/6)       . (1/6)
     = 2/36
```

3.8 Probability Laws

1. Probability of disjoint events

   A and B are said to be disjoint (as sets or conditions) iff.

   - $A \cap B = \varnothing$
   - $A \wedge B = false$

   If A and B are disjoint, then the probability of the union is the sum of probabilities:

   - If $A \wedge B = false$, then $P(A \vee B) = P(A) + P(B)$
   - If $A \cap B = \varnothing$, then $P(A \cup B) = P(A) + P(B)$

   Remark: do not confuse "disjoint" and "independent".

2. Probability of negation/complement

   - $P(\neg A) + P(A) = P(\neg A \vee A) = P(true) = 1$

   hence:

   - $P(\neg A) = 1 - P(A)$

   exercise: use sets instead of boolean expressions.

3. Law of total probability if B1, B2 disjoint and B1 $\vee$ B2 = true:

   - P(A) = P(A $\wedge$ B1) + P(A $\wedge$ B2)

   Indeed,

```
P(A ∧ B2) + P(A ∧ B1)
 = P((A ∧ B2) ∨ (A ∧ B1)) -- disjoint events
 = P(A ∧ (B2 ∨ B1))
 = P(A ∧ true)
 = P(A)
```

4. Probability of disjunction

   What if A and B are not disjoint?

   - P(A ∨ B) = P (A) + P(B) - P(A ∧ B)

```
Lemma: if A ⊆ B then P(A) + P(B\A) = P(B)
Proof: P(A) + P(B\A)
     = P (A ∪ (B\A))    -- disjoint events
     = P (B)

Proof of theorem:
  P(A ∪ B) = P(A ∪ (B\A))
           = P(A) + P(B\A))       -- disjoint events
           = P(A) + P(B) - P(A∩B) -- Lemma
```

5. Summary of Laws

   - $P(\Omega) = 1$
   - $P(\neg A) = 1 - P(A)$
   - $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
   - $P(A \wedge B) = P(A|B) \cdot P(B)$
   - if $B1, B2$ complementary, $P(A) = P(A \wedge B1) + P(A \wedge B2)$

3.9 Random variables with priors (discrete)

How to evaluate P(A), for an expression A depending on a random variable r?

Using the law of total probability:

```
P(A) = ∑(i:Ω) P(r=i ∧ A)
     = ∑(i:Ω) P(A | r=i) P(r=i)
```

We can even write

- $P(A) = \sum(i : \Omega)P(A[i/r])P(r = i)$

Writing A[i/r] to mean that we substitute r for i in the expression A. But A[i/r] no longer depends on a random variable. (It is either true or false). So it is less confusing to write Indicator instead of P, where Indicator(c) = 1 when c is true and 0 when c is false.

- $P(A) = \sum(i : \Omega)Indicator(A[i/r])P(r = i)$

In such an equation, we can call P(r=i) the prior probability of r=i.

1. Example (discrete) In the dice example, every time that we want to evaluate the probability of an event (or condition) A which **depends** on the roll of the dice, we can use the formula:

   - P(A) = $\sum$(i∈[1..6]) $\sum$(j∈[1..6]) P(A | d1 = i ∧ d2 = j) P(d1 = i ∧ d2 = j)

   If the dice are fair and independent, then P(d1 = i ∧ d2 = j) = 1/36, for any i,j., and we have:

   - P(A) = $\sum$(i∈[1..6]) $\sum$(j∈[1..6]) P(A | d1 = i ∧ d2 = j) / 36

   and even:

   - P(A) = $\sum$(i∈[1..6]) $\sum$(j∈[1..6]) Indicator(A[d1 = i,d2 = j]) / 36

   If the dice were unfair or dependent, we'd change the prior $P(d1 = i \wedge d2 = j)$ accordingly.

   Say if A is d1+d2 > 9:

   - P(d1+d2>9) = $\sum$(i∈[1..6]) $\sum$(j∈[1..6]) Indicator(i+j > 9) / 36

2. Random variables with priors (continuous) For continuous variables, we have:

   ```
   P(A) = ∫(x:Ω) f(r=i ∧ A) f_r(x) dx
        = ∫(x:Ω) Indicator(A[x/r]) f_r(x) dx
   ```

   with: f_r the PDF of the distribution of the random variable r.

   Example: probability that the coin falls on the table: Let

- $A \triangleq (x \in Table)$ (where Table is a subset of $\mathbb{R}^2$ representing the surface of the table.)
- $fCoin \triangleq 1/a$ (using a simple model where I can throw the coin anywhere in the room and a = room areaRoom area.)

```
P(A) = ∫(x:ℝ²) Indicator(x∈Table) 1/a dx
     = 1/a ∫(x:ℝ²) Indicator(x∈Table) dx
     = 1/a (∫(x∈Table) Indicator(x∈Table) dx + ∫(x∈(ℝ²\Table)) Indicator(x∈Table
     = 1/a (∫(x∈Table) 1 dx + ∫(x∈(ℝ²\Table)) 0 dx)
     = 1/a (1 ∫(x∈Table) dx + 0 ∫(x∈(ℝ²\Table)) dx)
     = 1/a (∫(x∈Table) dx)
     = 1/a t
     = t/a
```

Any idea of a better model? What would be the effect on the outcome?

3.10 Evidence and posteriors

Assume now that we have some **evidence** to account for.

In the case of the dice, we could somehow know that the sum is greater than 9. Then what is the **posterior** probability that the product is less than 20?

- $E \triangleq d1 + d2 > 9$
- $A \triangleq d1 \cdot d2 < 20$

We need to account for E:

- P(A | E) = P (A ∧ E) / P(E)

```
P(d1·d2 < 20 ∧ d1+d2 > 9) =
   ∑(i∈[1..6]) ∑(j∈[1..6]) Indicator(i+j > 9 ∧ i·j < 20) P(d1 = i ∧ d2 = j)
```

4 A Probabilistic Logic

The above gives an informal recipe to compute probabilities. It works for simple problems, but it's easy to make mistakes when tackling non-trivial problems. We set out to make the process systematic — and so it can also be the basis of algorithms. This systematic approach will help with the interpretation of natural langauge, which is our real goal.

Keeping a liberal mindset, we will call a possible world a (set of) priors, potentially modulo evidence. We will call such construction "spaces". On top of spaces we will define the (few) necessary concepts to interpret probabilistic syllogisms.

4.1 Probability distributions (1)

The basic building block to describe possible worlds (spaces) are probability distributions. We will list and discuss some of them.

- DiscreteUniform(Ω)
  - P(x) = $1/\#\Omega$ if x $\in \Omega$ = 0 otherwise
  - if all choices are equally probable — for a finite set of events.
- Uniform(a,b)
  - PDF(x) = 1/(b-a) if x $\in$ [a,b] = 0 otherwise
  - if all choices are equally probable — if the set of events is continuous and bounded.
- Bernoulli(p)
  - P(0) = 1-p
  - P(1) = p
  - Two choices, which are not necessarily equally probable.
  - In our example, we can represent the space of Balls by Bernoulli(ρ)

```
Ball = Bernoulli(ρ)
```

4.2 Probability distributions (2)

- Normal(μ,σ)
  - PDF = $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
  - Often used to model a random variable which depends itself on many variables in an unknown way
  - mean = μ
- Beta(α,β), α,$\beta > 0$

- PDF(x) = $\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$ if x ∈ [0,1], 0 otherwise
  where $\mathrm{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and Γ is the Gamma function.
- Useful to model bounded variables, with non-uniform distributions.
- Beta(1,1) = Uniform[0,1]
- Mean = $\alpha/(\alpha+\beta)$
- increasing α "pushes" the distribution towards 1; β towards 0.

4.3 Cartesian products (×) of spaces

If A and B are spaces, then A × B is a space.

Example: Uniform [0,1] × Uniform [0,1]

If x is taken in A × B, then we call $\pi_0(x)$ the first component of x, which lives in space A, and $\pi_1(x)$ its second component, living in space B.

The exponential notation is often used, defined by induction on the exponent:

- $A^{n+1} = A \times A^n$
- $A^1 = A$

Most models will involve more than a single variable, and thus involve cartesian products in a way or another.

In our motivational example, if B is the space describing one ball, then B^5 is the distribution of five (independent!) balls.

4.4 Σ-spaces

A generalisation of cartesian product, where the 2nd space can depend on the first one.

If A is a space and B is a space, Σ(x:A)B is a space. Additionally, the variable x can occur in B.

Examples:

```
A₀ = Σ(lo:Uniform[0,1]) Uniform[lo,2]
A = Σ(α:Uniform[2,5]) Σ(β:Uniform[2,5]) Beta(α,β)
```

For consistency, from now on if we write x:A, then x is a random variable taken in space A.

We can use projections as before to extract the components. Attention:

- If $z : (\Sigma(x : A)B)$, then $\pi_1(z) : B[\pi_0(z)/x]$

It is convenient to use record notation for Σ types. The space below is isomorphic to the above example:

```
A = [α:Uniform[2,5];
     β:Uniform[2,5];
     x:Beta(α,β)]
```

Additionally if $z : A$ then $z.x : Beta(z.\alpha, z.\beta)$

4.5 Filtering probability distributions: IsTrue(ϕ)

To represent evidence, we introduce the space IsTrue(ϕ), where ϕ is a Boolean-valued expression. If ϕ is true, then IsTrue(ϕ) has a single element, which we will call $\diamond$, by convention.

Hence we could say:

- P($\diamond$) = 1

But we will never manipulate such probability directly any more.

- The space IsTrue(ϕ) is *empty* if ϕ is false.

This is useful to filter out impossible worlds in combination with Σ:

- $\Sigma(lo : Uniform[0,1])\Sigma(hi : Uniform[0,1])(IsTrue(lo < hi)) \times Uniform[lo, hi]$

We may sometimes omit IsTrue altogether and simply write the following for the same space:

- $\Sigma(lo : Uniform[0,1])\Sigma(hi : Uniform[0,1])(lo < hi) \times Uniform[lo, hi]$

In record notation:

```
[lo:Uniform[0,1];
 hi:Uniform[0,1];
 lo<hi;
 x:Uniform[lo,hi]]
```

IsTrue allows us to model any dependency among random variables.

1. Example In our running example, if a the space of a blue ball can be written as:

   $BlueBall = \Sigma(b : Ball).IsTrue(b = blue)$

   (remember $Ball = Bernoulli(\rho)$)

4.6 Expected truth value (aka "Probability")

We can now conveniently phrase our problems in this framework:
If we let Die = DiscreteUniform({1,2,3,4,5,6})

```
Ω =  [d1:Die;
      d2:Die;
      d1+d2 > 9]
```

Given a random ω:Ω, we'd be interested in the truth value of

- $\phi = \omega.\text{d1} \cdot \omega.\text{d2} > 20$

But ϕ depends on which world ω we choose. So the *probability* of ϕ is given by the expected value of its indicator function.
Hence we define:

- Pb (ω:Ω) ϕ = E[ω:Ω] (Indicator(ϕ))

In reality, $Pb(x : A)C$ is the proportion of the space A satisfying the condition C. If A is an accurate model of all possible worlds, then this number commonly known as the "probability of C".
(We will see later how to evaluate expected values.)

1. Remark

   $Pb(z : \Sigma(x : A)IsTrue(B))C \neq Pb(x : A)(B \to C)$

   On the lhs, if B is false x is not counted. On the rhs if B is false it is counted as satisfying the condition.

2. Example

   If I throw a coin, what is the probability that it falls on the table? (But now with a better model. . .)

   Possible worlds:

- $\Omega = Normal(Aim, Error)^2$

Note that Normal is just a choice! we could use anything else.

Answer:

- $Pb(X : \Omega)(X \in Table)$

4.7 Answer To Introductory Problem

link back to the introductory problem

One might think that a simple answer is "3/4". But is this correct? Let's try to use the concepts developped so far.

We start by describing the space of possible worlds:

```
Ball = Bernoulli(ρ)
BlueBall = Σ(b:Ball) IsTrue(b=0)
RedBall  = Σ(b:Ball) IsTrue(b=1)

Ω = [ρ : Uniform [0,1];  -- ??? what would you pick here
     ball1 : RedBall;    -- note this depends on ρ
     ball2 : RedBall;
     ball3 : RedBall;
     ball4 : BlueBall;
     ball5 : Ball]
```

What we want to evaluate is the probability to we have a red ball as a 5th pick:

- $\theta = Pb(w : \Omega)(w.ball5.b = 1)$

What do you think is the value of θ? This is the ratio of the "size" of spaces. But how do we evaluate the "size" of spaces? This is the topic of the rest of the lecture. (After we show some more examples...)

4.8 Example: drug test

Suppose that a test for using a particular drug is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. What is the probability that a randomly selected individual with a positive test is a drug user?

1. Solution

```
Ω = [isUser : Bernoulli(0.005)
    ;testPositive : Bernoulli(if isUser then 0.99 else 0.01)
    ;IsTrue(testPositive)]

Pb(ω:Ω) (ω.isUser)
```

4.9 Example: betting on games

- Consider the game "Sloubi".
- Each player p of Sloubi is assigned a rating ρ_p. The rating is intrinsic to each player, and never changes.
- There is an element of randomness in Sloubi. In any match, p will win over q if $\rho_p > \rho_q + m$, with m taken in Normal(0,100). (Even worse players will win, sometimes.)
- Alice wins over Bob, Bob wins over Charles and David. What is Alice's probability to win over David in their next game?

1. Solution

```
Player = Normal(1000,500) -- (or something else! -- this was missing from the ques
Match = Normal(0,100)

win(m,p,q) = p+m > q

Ω = [alice:Player;
     bob:Player;
     charles:Player;
     david:Player;
     m1,m2,m3,m4:Match;
     win(m1,alice,bob);
     win(m2,bob,charles);
     win(m3,bob,david)]

Answer = Pb (ω:Ω) (win(ω.m4,ω.alice,ω.david))
```

4.10 Meaning of probabilistic logic expressions

1. Spaces

   We define the integrator of f over A ($[\![A]\!]f$) as a generalisation of the integration/summation of f(x) for the possible elements x of A, multiplied by the density of elements of A.

   We define:

$$
\begin{aligned}
[\![\mathsf{Distr}(d)]\!]f &= \int_{\mathbb{R}} \mathrm{PDF}_d(x) \cdot f(x) dx \\
[\![\mathsf{Distr}(d)]\!]f &= \sum P_d(x) \cdot f(x) \\
[\![\mathsf{IsTrue}(\phi)]\!]f &= Indicator([\![\phi]\!]) \cdot f(\diamond) \\
[\![\Sigma(x : A)B]\!]f &= [\![A]\!](\lambda x.[\![B]\!](\lambda y.f(x, y)))
\end{aligned}
$$

2. Measure To compute the measure of a space, we integrate over the whole space the constant value 1 — thus "counting" every element with the same unit weight.

   $\mathsf{measure}(A) = [\![A]\!](\lambda x.1)$

3. Expressions Expressions are evaluated structurally:

$$
\begin{aligned}
[\![x]\!] &= x \\
[\![c]\!] &= c \\
[\![e_1 + e_2]\!] &= [\![e_1]\!] + [\![e_2]\!] \\
[\![e_1 \cdot e_2]\!] &= [\![e_1]\!] \cdot [\![e_2]\!] \\
[\![\pi_i(e)]\!] &= \pi_i [\![e]\!] \\
[\![(t, u)]\!] &= ([\![t]\!], [\![u]\!])
\end{aligned}
$$

4. Propositions Propositions are evaluated structurally:

$$
\begin{aligned}
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \wedge [\![\psi]\!] \\
[\![e_1 < e_2]\!] &= [\![e_1]\!] < [\![e_2]\!]
\end{aligned}
$$

5. Lemma: integrators are linear operators

   Lemma:

   - $[\![A]\!](\lambda x.kf(x)) = k[\![A]\!]f$
   - $[\![A]\!](\lambda x.f(x) + g(x)) = [\![A]\!]f + [\![A]\!]g$

   Proof: By induction on A, relying on the linearity of $\sum$ and $\int$.

6. Properties of measures

$$\begin{aligned}
\mathsf{measure}(\mathsf{Distr}(d)) &= 1 \\
\mathsf{measure}(\mathsf{IsTrue}(\phi)) &= Indicator[\![\phi]\!] \\
\mathsf{measure}(A \times B) &= \mathsf{measure}(A) \times \mathsf{measure}(B) \\
\mathsf{measure}(\Sigma(x : A)B) &= [\![A]\!](\lambda x.\mathsf{measure}(B))
\end{aligned}$$

   - Proposition: For every space A, $\mathsf{measure}(A) \leq 1$.
   - Lemma: For every space A, $[\![A]\!]f \leq max\,\{fx \mid x \in A\} \cdot \mathsf{measure}(A)$.

7. Expected value

   The expected value of f over A is defined as follows:
   $E_{x:A}[f(x)] = \frac{[\![A]\!]f}{\mathsf{measure}(A)}$

8. Expected truth value

   (a.k.a. "probability")

   The expected truth value of ϕ over Ω is defined as follows:
   $Pb(\omega : \Omega)(\phi) = E_{\omega:\Omega}(Indicator(\phi)) = \frac{[\![A]\!]\lambda x.Indicator(\phi[x])}{\mathsf{measure}(A)}$

9. Example: Drug test Given:

```
Ω = [isUser : Bernoulli(0.005)
    ;testPositive : Bernoulli(if isUser then 0.99 else 0.01)
    ;IsTrue(testPositive)]
```

   - Compute: $E[\omega : \Omega](Indicator(\omega.isUser))$
   - Answer: $([\![\Omega]\!]\lambda\omega.Indicator(\omega.isUser))/measure(\Omega)$

   Compute now the denominator:

```
measure(Ω)
=
Σ(isUser:Bool) Bernoulli(0.005)(isUser) ·
Σ(testPositive:Bool) Bernoulli(if isUser then 0.99 else 0.01)(testPositive) ·
Indicator(testPositive)
=
Σ(isUser:Bool) Bernoulli(0.005)(isUser) ·
Bernoulli(if isUser then 0.99 else 0.01)(true) ·
=
Σ(isUser:Bool) Bernoulli(0.005)(isUser) ·
if isUser then 0.99 else 0.01 ·
=
Bernoulli(0.005)(false) (if false then 0.99 else 0.01) +
Bernoulli(0.005)(true) (if true then 0.99 else 0.01)
=
0.995 0.01 + 0.005 0.99
=
0.0149
```

Compute the numerator

```
⟦Ω⟧ λω. Indicator(ω.isUser)
=
Σ(isUser:Bool) Bernoulli(0.005)(isUser) ·
Σ(testPositive:Bool) Bernoulli(if isUser then 0.99 else 0.01)(testPositive) ·
Indicator(testPositive) ·
Indicator(isUser)
=
Bernoulli(0.005)(true) ·
Bernoulli(if true then 0.99 else 0.01)(true)
=
0.005 ·
0.99
=
0.00495
```

So the ratio is: 0.332214765101

4.11 Exercise: Evaluating the introductory problem

```
ok =
```

```
∫(ρ:[0..1]) dρ
∑(b1:[0,1]) Bernoulli(ρ,b1) ·
∑(b2:[0,1]) Bernoulli(ρ,b2) ·
∑(b3:[0,1]) Bernoulli(ρ,b3) ·
∑(b4:[0,1]) Bernoulli(ρ,b4) ·
∑(b5:[0,1]) Bernoulli(ρ,b5) ·
(b1 · b2 · b3 · (1-b4) · b5)

total =
∫(ρ:[0..1]) dρ
∑(b1:[0,1]) Bernoulli(ρ,b1) ·
∑(b2:[0,1]) Bernoulli(ρ,b2) ·
∑(b3:[0,1]) Bernoulli(ρ,b3) ·
∑(b4:[0,1]) Bernoulli(ρ,b4) ·
(b1 · b2 · b3 · (1-b4))
```

- θ = ok/total

Reminder:

- Bernoulli(ρ,0) = ρ
- Bernoulli(ρ,1) = 1-ρ

4.12 Evaluation using Probabilistic Programming

Spaces as we have defined can also be written as *functions* which generate a *sample* in the space of interest.

Following this idea, our running example would be written this way:

```
example = do
  p <- sample (Uniform 0 1)
  let ball = sample (Bernoulli p)
      redBall = do
        b <- ball
        observe b
      blueBall = do
        b <- ball
        observe (not b)
  redBall
```

```
redBall
redBall
blueBall
ball5 <- ball
return (ball5)
```

1. Running Link to the program

4.13 Final note on Beta distribution.

If we have n reds and m blues, Beta(n+1/2, m+1/2) is the expected distribution of the proportion of red balls.

In particular, the expected value of this proportion is (n+0.5) / (n+m+1)

Therefore, in our example, we expect to see not a 3/4 prediction for the ratio of red ball, but rather but 3.5/5. (Which is exactly what the program predicts!)

4.14 SKIP probability density/mass functions

We can define a generic notion of probability distribution over the spaces defined as above.

Let's first define G[A](x,y) such that G[A](x,y) = 1 if x = y, 0 otherwise.

By induction:

$$\begin{aligned} G[\mathsf{Distr}(d)](x,y) &= \delta(x-y) \\ G[\mathsf{Distr}(d)](x,y) &= Indicator(x=y) \\ G[\mathsf{IsTrue}(\phi)](x,y) &= 1 \\ G[\Sigma(z:A)B]((x,y),(x',y')) &= G[A](x,x') \cdot G[B](y,y') \end{aligned}$$

Then the Probability (mass) distribution over A is given by:

$P_A(x) = \frac{1}{\mathsf{measure}(A)}[\![A]\!](\lambda y.G[A](x,y))$

Note that if A is continuous, he argument of $[\![A]\!]$ is integrated, so δ always occurs under an integral.

5 A Probabilistic model of natural language

5.1 Montegovian semantics

As a quick reminder, we can associate types with syntactic categories, in the following manner:

```
type Pred = Ind → Prop
type CN = Ind → Prop
type VP = Ind → Prop
type NP = VP → Prop
type Quant = CN → NP
type Ind = ...
```

But what are individuals? It is mysterious!

In fact, Montegovian semantics consider Individuals to be *abstract*. This means that nothing needs to be known about them to be able to interpret phrases. However, if one needs to give specific semantics to lexical items (perhaps in specific domains), we need to get more concrete.

In fact we keep (nearly) all Montegovian semantics, and make certain things concrete.

5.2 Context: (natural) language inference

Classic syllogism:

- all men are mortal
- socrates is a man
- socrates is mortal?

It can be intepreted in probabilistic logic this way:

```
Ω = [man : CN;
     mortal : VP;
     socrates : Ind;
     man(socrates);
     ∀(x:man) mortal(x)
     ]

Pb (ω:Ω) (ω.mortal (ω.socrates))
```

If the definitions of CN,VP,Ind, etc. are well-chosen we expect the above proportion to evaluate to 1.

Note that we quantify man over all CNs; so we do not have any *a-priori* notion for what "man" means — we sample over the whole space of CNs. The distribution for "man" gets refined by evidence (in this case "∀(x:man) mortal(x)", "man(socrates)").

Indeed, the semantics for spaces that we gave mean that all worlds where we can find non mortal men will be filtered out.

The goal is to

- interpret each syntactic category as a space
- interpret each syntactic operator as a function from/to the apropriate spaces.

5.3 Propositions

We'll simply interpret propositions as Boolean-valued expressions.

```
type Prop = Bool
```

5.4 Individuals

Fortunately we now have a way to interpret individuals as elements in a space.

Examples:

- multi-variate normal distribution of dimension k
  - covariance matrix (?)
- uniform distribution in a box $[0..1]^k$

```
Ind = Normal(0,1)^n
Ind = Uniform(0,1)^n
```

- Discussion: what would *you* choose? Why?

This idea is directly inspired from machine learning: individual (situations) can be represented by a vector.

This is indeed used for:

- Words
- Sentences
- Images

5.5 Reminder: set cardinalities.

If #A = n, then #(A → Bool) = 2^#A. So, there are "exponentially many" more predicates over a set than there are elements in the set.

A related fact is that $\mathbb{N}$ is countable, but the set of predicates over natural numbers $\mathbb{N} \to$ Bool is uncountable.

There is an obvious way to integrate over $[0, 1]$, but how to integrate over $[0, 1] \to \mathit{Bool}$? How to take "the average" over all possible predicates?

5.6 Space of predicates

We're deliberately going to restrict the set of possible predicates to make our endeavour possible. Hopefully, it's enough to limit oneself to a small enough (samplable) subset and still have a useful model.

If words can be represented by a vector, then so can predicates (hopefully).

(NOTE: other ideas would be to sample from a set of programs which implement predicates.)

1. Morphing spaces

   The idea is to map vectors to predicates. How to do this? We need to extend our language of spaces with the construction {e | x : A}, for any space A, with the semantics:

   - ⟦{e[x] | x : A}⟧ f = ⟦A⟧ (λx. f(e[x]))

2. Idea 1 If an individual is represented by a vector x and a vector p represents a predicate, then x is said to satisfy the predicate if $p \bullet x > 0$. (Ie, both vector are oriented in the same direction in the underlying euclidean space.)

   - Pred = {λx. p • x > 0 | p : Normal(0,1)^n};

3. Idea 2

   If an individual is represented by a vector x and a pair of vectors p, q represent a predicate, then x is said to satisfy the predicate if x is in the box delimited by the corners p and q.

   - Pred = {λx. ∀i. $(\text{b.p})_i < \text{x}_i < (\text{b.q})_i$ | b : [p : Uniform(0,1)^n, q : Uniform(0,1)^n]}

   Exercise: check that Pred is a subspace of Ind → Bool.

5.7 Common nouns

We can interpret common nouns as spaces.

$[\![\text{cn}]\!] = \Sigma(\text{x:Ind})([\![\text{cn}]\!]\ \text{x})$

Ideally we'll have a coercion from $\Sigma(\text{x:Ind})([\![\text{cn}]\!]\ \text{x})$ to Ind. One simple way to deal with this is to use records. Indeed:

- $[i : Ind]$ is a subtype of $[i : Ind; cni]$

and there is an implicit subtyping coercion.

5.8 Generalised quantifiers

Thanks to the probabilistic logic, we can interpret generalized quantifiers. (Most, Few, etc.)

First we add two new constructions (for propositions in probabilistic logic).

- $[\![\text{AtLeast } \theta \text{ (x:A). } \phi]\!] = \text{measure } (\Sigma(\text{x:A})\phi) > \theta \text{ measure(A)}$
- $[\![\text{AtMost } \theta \text{ (x:A). } \phi]\!] = \text{measure } (\Sigma(\text{x:A})\phi) < \theta \text{ measure(A)}$

And:

- $[\![\text{Most cn vp}]\!] = \text{AtLeast } \theta \text{ (x : } [\![\text{cn}]\!]) \text{ } ([\![\text{vp}]\!]\ \text{x})$
- $[\![\text{Few cn vp}]\!] = \text{AtMost (1-}\theta\text{) (x : } [\![\text{cn}]\!]) \text{ } ([\![\text{vp}]\!]\ \text{x})$

(Attention: the bracket notation is overloaded.)

1. Example:

   - most men are mortal
   - socrates is a man
   - socrates is mortal?

   World:

```
Ω = [man : CN;
     mortal : VP;
     socrates : Ind;
     man(socrates);
     AtLeast θ (x:man) mortal(x)
     ]
```

- Compute: $Pb(\omega : \Omega)(\omega.mortal(\omega.socrates))$

link to implementation

2. Choice of θ

   One can choose θ by studying native speakers. However, one should expect that you won't get a single value of θ which will fit all situations, but rather you'll observe a distribution for θ. Probabilistic logic is ideally suited to deal with this.

```
Ω = [θ : Beta(5,2); -- for example
     man : CN;
     mortal : VP;
     socrates : Ind;
     man(socrates);
     AtLeast θ (x:man) mortal(x)
     ]
```

   Below we'll leave θ abstract.

3. Exercise: Show that Few and Most are dual

   Example : few men live to 100 years = most men do not live to 100 years.

   Theorem:

   - ⟦⟦Few cn vp⟧⟧ = ⟦⟦Most cn (don't vp)⟧⟧

   Proof:

```
  ⟦⟦Few cn vp⟧⟧
= ⟦AtMost (1-θ) (x : ⟦cn⟧) (⟦vp⟧ x)⟧                          -- def.
= measure (Σ(x:⟦cn⟧)(⟦vp⟧ x)) < (1-θ) measure(⟦cn⟧)          -- def.
= ⟦⟦cn⟧⟧ λx → Indicator(⟦vp⟧ x)  < (1-θ) measure(⟦cn⟧)     -- def.

⟦⟦Most cn (don't vp)⟧⟧
= ⟦AtLeast θ (x:⟦cn⟧)(⟦don't vp⟧ x)⟧                                  --
= ⟦AtLeast θ (x:⟦cn⟧)(¬ ⟦vp⟧ x)⟧                                     --
= measure (Σ(x:⟦cn⟧)(¬⟦vp⟧ x)) > θ measure(⟦cn⟧)                     --
= ⟦⟦cn⟧⟧ λx → Indicator(¬⟦vp⟧ x)  > θ measure(⟦cn⟧)                 --
= ⟦⟦cn⟧⟧ λx → (1 - Indicator(⟦vp⟧ x))  > θ measure(⟦cn⟧)            --
```

```
= ([[[cn]]] λx → 1) - ([[[cn]]] λx → Indicator([[vp]] x))  > θ measure([[cn]])      -- in
= measure([[cn]]) - ([[[cn]]] λx → Indicator([[vp]] x))  > θ measure([[cn]])       -- de
= ([[[cn]]] λx → Indicator([[vp]] x)) - measure([[cn]])   < -θ measure([[cn]])     -- ne
= ([[[cn]]] λx → Indicator([[vp]] x))  <  measure([[cn]]) - θ measure([[cn]])      -- ad
= ([[[cn]]] λx → Indicator([[vp]] x))  <  (1-θ) measure([[cn]])                    -- f
```

5.9 Universal Quantifiers

We add the construction $\forall x : A.\phi$ to propositions in probabilistic logic, with the following meaning:

- $[\![\forall x : A.\phi]\!] = measure A \leq measure(\Sigma(x : A)\phi)$

(With the idea that have as many individuals in A as in $\Sigma(x : A)\phi$)
The following definition is also possible:

- $\mathsf{measure}(\Sigma(x : A)(\neg\phi)) = 0$

If we have in mind a logic with excluded middle.
Exercise: why would this definition be equivalent?
Given the above, we can interpret natural language phrases such as "every man is mortal", as follows:

- $[\![Everycnvp]\!] = \forall x : [\![cn]\!].[\![vp]\!](x)$

1. Pitfalls

   Assume

   - A = [-1..1]
   - ϕ = (x $\neq$ 0)

   We have:

   - measure(A) = 2
   - measure(Σ(x:A)ϕ) = 2

   Indeed, we filtered out a single point — its measure is 0

   And according to the above definition:

   - $[\![\forall$x:A. $\phi]\!]$ = true

(So this operator really means "for almost all" in probabilistic logic)

2. Dealing with this pitfall

   - attempt to have a precise measure that counts single elements
     - not computable, because HOL is undecidable
   - use "soft transitions"
     - still does not make ∀x:A. ϕ coincide with the usual definition (but can help with the approximation algorithms in many cases.)
   - do not use problematic domains
     - unless otherwise note, this is what we will do.

5.10 Existential Quantifiers

One can define existential quantifiers by dualizing universals in either version.

$[\![\exists$x:A. $\phi]\!] = 0 <$ measure $(\Sigma$(x:A)$\phi)$

5.11 Comparatives

Mary is tall John is tall

"Mary is taller than John"?

We can support scalar predicates and comparatives, by generalizing predicates.

Scalar is a subspace of functions from individuals to reals, such that if the function evaluates to a positive value for individual x, then x is considered to satisfy the non-scalar retraction of the predicate.

+BEGIN$_{\text{SRC}}$ haskell is :: Scalar → Ind → Prop is m x y = m x > 0 #+END$_{\text{SRC}}$

But then one can compare individuals with respect to any scalar predicate:

```
more :: Scalar → Ind → Ind → Prop
more m x y = m x > my
```

1. Idea 1 The expression $b + d \cdot x$ can be interpreted as a degree to which the individual x satisfies the property characterised by (b, d). Thus satisfying a scalar predicate is

   - Scalar = {λx. p • x | p : Normal(0,1)^n}

Scalars can be converted to Pred in the obvious way.

- Scalar = $\{\lambda$x. $\forall$i. $(\text{b.p})_i < \text{x}_i < (\text{b.q})_i \mid$ b : [p : Uniform(0,1)^n, q : Uniform(0,1)^n]$\}$

2. Idea 2 The degree to which an individual x satisfies a property characterised by a box centered at c and of dimensions d is given by $s(x)$, with:

   - $s(x) = 1 - \max\left\{ \frac{\mathsf{abs}(x_i - c_i)}{d_i} \;\middle|\; i \in [1..n] \right\}$

   This definition entails that the subspace corresponding to a predicate coincides with the space where its degree of satisfaction is positive.

   Remark: $s(x) > 0$ iff. x is inside the box.

3. Example

   ```
   Ω = [ tall : Scalar
       ; john : Ind
       ; mary : Ind
       ; more tall john mary
       ]

   Pb (ω:Ω) (is tall john) > 0.5
   ```

   That is, if we observe that “John is taller than Mary", we will infer that “John is tall" is slightly more probable than “John is not tall".

   link to implementation

5.12 Subsective Graded Adjectives

How to make the following consistent?

- Dumbo is not a large elephant
- Mickey is a large mouse
- Dumbo is larger than Mickey

Use a definition of "is" which is relative to the class (cn) in question:

```
is :: Scalar → Ind → CN → Prop
is m x cn = m x > E(z:cn)[m z]
```

5.13 TODO units of measures

6 Implementation aspects

So far, we can:

- evaluate natural language probabilistic syllogisms to probabilistic logic propositions
- evaluate such probabilities of such propositions as mathematical formulas

BUT:

- these formulas involve integrals which are typically not easy to compute (when involving non-trivial spaces).

Fortunately there are ways to approximate probabilities directly without resorting to symbolic integration.

6.1 Markov Chain Monte Carlo

1. Monte Carlo methods:

   To evaluate Pb (x:A) C:

   - take a random x:A
   - check if C(x) holds, increment q.

   After sufficiently many trials:

   - Pb (x:A) C $\approx$ q/n

2. Markov Chain

   - Informally: "a random walk"
   - assume set of states S, and a starting state s_0.
   - for each pair of states (s,t), assume a probability P(s,t) to transition from s to t.
   - at each step, transition from a state to another according to the given probabilities.
   - an interesting question: after an infinite number of steps, what is the probability to end at a given state s_f?

3. MCMC

   Sampling in a complicated space A is not so easy. Typically we have a space with many dimensions (cartesian product of many distributions), and a complicated filtering function (ϕ is complex).

   One way to solve this method is to **define** a Markov Chain where:

   - each state is an element of the space
   - the transition function is such that, as much as possible, s(x,y) > s(x,y') if P_A(y) > P_A(y').
   - note that this is not an easy thing to define when there are many variables (also, with Σ/IsTrue the existence of variables depend on the value of others.)

   We use this kind of random walk to sample in A, and apply the monte carlo method as usual to evaluate the proportion.

   Potential issues:

   - You never find a valid (x:A) to start with
   - The space is divided in regions which are not connected, or a walk from one to the other is highly improbable.

6.2 Choice of priors

In any $\Sigma(x : A)\phi$ a potential pitfall is to chose A very wide. (eg. a (cartesian product of) uniform distribution with large support), followed by a very restrictive ϕ. In such situation the Monte Carlo algorithm will spend a lot of time sampling elements of A only to discard them. It is better to restrict A to one of its subspace B so that ϕ becomes more easy to satisfy on B.

6.3 Inner evaluation of proportions

When using quantifiers, we evaluate more proportions/measures, and an inner instance of the MCMC algorithm must be employed. This can be very slow! A potential way out: when we use boxes some integrals can be computed symbolically and we save much resources.

6.4 More methods

- variational
- and even others

6.5 Choice of prior (again)

The (Uniform 0 1) prior is biasing the result towards 1/2. In order not to bias the result, one should use the Jeffrey's prior, which in this case is Beta(0.5, 0.5).

The Jeffrey's prior is given by the square root (the determinant of) the Fisher information (matrix) I. (I = variance of the derivative of log of density.)

6.6 Haskell

Example implementation of MCMC in Haskell

Link

6.7 More Probabilistic programming packages

- STAN: `https://mc-stan.org/`
- WebPPL `http://dippl.org/`

7 References

- Variational Inference: A Review for Statisticians `https://arxiv.org/pdf/1601.00670.pdf`
- A Tutorial on Variational Bayesian Inference `http://www.orchid.ac.uk/eprints/40/1/fox_vbtut.pdf`
- Bayesian inference `https://en.wikipedia.org/wiki/Bayesian_inference#Bayesian_inference`