Mathematical Foundations of Plant Semantics

Simon Castellan\textsuperscript{1}, Jos Käfer\textsuperscript{2}, Eric Tannier\textsuperscript{3}

\textsuperscript{1} Inria Rennes
\textsuperscript{2} CNRS
\textsuperscript{3} Inria Lyon

February 16th, 2023
Another research

Can we do research that:

- solve a real need arising from society,
- empowers users without enslaving them (Illich’s *conviviality*),
- is somewhat aligned with the ecological transition.
Another research

Can we do research that:

▶ solve a real need arising from society,
▶ empowers users without enslaving them (Illich’s *conviviality*),
▶ is somewhat aligned with the ecological transition.

A case study, Pl@ntNet: can we improve on the points above?

▶ Hard to trust
▶ Very knowledgeable but a bad teacher
▶ What if it goes away?
Flore complète de la France et de la Suisse, pour trouver facilement les noms de plantes, SANS MOTS TECHNIQUES
The rabbithole

*Can we automate the work of building determination keys and alleviate some of their limits?*

**Output:** offline app or even paper version of the key.

**Challenges:**

- No open source morphological database.
- No formal description of plants.
The rabbithole

*Can we automate the work of building determination keys and alleviate some of their limits?*

**Output**: offline app or even paper version of the key.

**Challenges**:

- No open source morphological database.
- No formal description of plants.
- Avoid over-engineering!
- Participative research to build data.
The ID3 algorithm [Quinlan’86]

In what order should we ask the questions?

1. Start with an initial probability distribution $d$.
2. For every question $q$:
   - Compute the average information after $q$.
3. Ask the question with the largest information.
4. Update $d$ with the user answer and go back to (1).

Greedy and non-optimal, but good enough for now.
The ID3 algorithm [Quinlan’86]

In what order should we ask the questions?

**Bayesian algorithm** using **information theory**:

1. Start with an **initial probability distribution** $d$. 
The ID3 algorithm [Quinlan’86]

In what order should we ask the questions?

**Bayesian algorithm** using **information theory**:

1. Start with an **initial probability distribution** $d$.
2. For every question $q$:
   - compute the average **information** after $q$.
3. Ask the question with the **largest information**.
The ID3 algorithm [Quinlan’86]

In what order should we ask the questions?

**Bayesian algorithm** using **information theory**:

1. Start with an **initial probability distribution** $d$.
2. For every question $q$:
   - compute the average **information** after $q$.
3. Ask the question with the **largest information**.
4. **Update** $d$ with the user answer and go back to (1).
The ID3 algorithm [Quinlan’86]

In what order should we ask the questions?

**Bayesian algorithm using information theory:**

1. Start with an initial probability distribution \( d \).
2. For every question \( q \):
   - compute the average information after \( q \).
3. Ask the question with the largest information.
4. **Update** \( d \) with the user answer and go back to (1).

Greedy and non-optimal, but good enough for now.
An aside about information theory [Shannon’48]

Information is usually defined as the opposite of entropy:

$$\text{entropy} : \mathcal{D}(X) \to \mathbb{R}$$
$$d \mapsto -\sum_{x \in X} d(x) \times \log(d(x))$$
An aside about information theory [Shannon’48]

Information is usually defined as the opposite of entropy:

\[
\text{entropy} : \mathcal{D}(X) \to \mathbb{R} \\
\quad d \mapsto -\sum_{x \in X} d(x) \times \log(d(x))
\]

Extremal values:

\[
\begin{align*}
\text{entropy}(\text{uniform}(\{1, \ldots, n\})) &= \log(n) \\
\text{entropy}(\text{dirac}(x)) &= 0
\end{align*}
\]
An aside about information theory [Shannon’48]

**Information** is usually defined as the opposite of **entropy**:

\[
\text{entropy} : \mathcal{D}(X) \rightarrow \mathbb{R} \\
\quad d \mapsto -\sum_{x \in X} d(x) \times \log(d(x))
\]

Extremal values:

\[
\begin{align*}
\text{entropy}(\text{uniform}\{1, \ldots, n\}) &= \log(n) \\
\text{entropy}(\text{dirac}(x)) &= 0
\end{align*}
\]

**Maximizing information \(\leftrightarrow\) minimizing entropy**
Bayesian Update

Input:
- $d \in \mathcal{D}(\text{Species})$: prior knowledge of what species it may be.
- $o \in \text{Obs}$: user answer ("red flower")

Output:
- $d' \in \mathcal{D}(\text{Species})$: posterior knowledge

For $s \in \text{Species}$:

\[
d'(s) \propto d(s) \times \text{score}(s, o)
\]

\[
P(\text{we see } s | \text{observing } o)
\]

\[
P(\text{we see } s)
\]

\[
P(\text{observing } o | \text{we see } s)
\]
Bayesian Update

Input:

- $d \in \mathcal{D}(\text{Species})$: prior knowledge of what species it may be.
- $o \in \text{Obs}$: user answer ("red flower")

Output:

- $d' \in \mathcal{D}(\text{Species})$: posterior knowledge

For $s \in \text{Species}$: (Bayes’ Law)

$$d'(s) \propto d(s) \times \text{score}(s, o)$$

$$P(\text{we see } s \mid \text{observing } o) \quad P(\text{we see } s) \quad P(\text{observing } o \mid \text{we see } s)$$
Input to the ID3 algorithm

(* Representation of the description of a species *)
(* e.g. "white flowers, simple lanceolate leaves" *)

`type species`

(* Representation of observation (user answers). *)

`type observation`

(* Given an observation, how likely is it to be a particular species? *)

`val score : species -> observation -> float`

How to describe species and observation?

⇝ `score` tangles species and observation.
General shape of model

We distinguish two things:

- **Plant description**, which are certain and exhaustive
  “Tree with white flowers, large leaves ...“

- **Observation**, which are uncertain
  “Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)”
We distinguish two things:

- **Plant**: *Plant description*, which are certain and exhaustive
  
  “Tree with white flowers, large leaves ...“

- **Obs**: *Observation*, which are uncertain
  
  “Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)”

We let $\text{Obs} = \mathcal{D}(\text{Plant})$. 
General shape of model

We distinguish two things:

- **Plant**: *Plant description*, which are certain and exhaustive
  “Tree with white flowers, large leaves ...“

- **Obs**: *Observation*, which are uncertain
  “Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)”

We let $\text{Obs} = \mathcal{D}(\text{Plant})$.

$$P(\text{observing } o \mid \text{we see } s)$$
General shape of model

We distinguish two things:

▶ **Plant**: **Plant description**, which are certain and exhaustive
  “Tree with white flowers, large leaves ...“

▶ **Obs**: **Observation**, which are uncertain
  “Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)”

We let \( \text{Obs} = \mathcal{D} (\text{Plant}) \).

\[
P(\text{observing } o \mid \text{ we see } s) \\
= \sum_{p \in \text{Plant}} o(p) \times P(\text{observed plant is } p \mid \text{ we see } s)
\]
We distinguish two things:

- **Plant**: Plant description, which are certain and exhaustive
  
  “Tree with white flowers, large leaves …“

- **Obs**: Observation, which are uncertain

  “Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)”

We let $\text{Obs} = \mathcal{D}(\text{Plant})$.

\[
P(\text{observing } o \mid \text{ we see } s) = \sum_{p \in \text{Plant}} o(p) \times P(\text{observed plant is } p \mid \text{ we see } s)
\]

Thus we can also let Species $= \mathcal{D}(\text{Plant})$!
General shape of model

We distinguish two things:

- **Plant**: **Plant description**, which are certain and exhaustive
  
  "Tree with white flowers, large leaves . . ."

- **Obs**: **Observation**, which are uncertain
  
  "Tree with whiteish flowers (maybe rose), not sure about leaves (its winter)"

We let $\text{Obs} = \mathcal{D}(\text{Plant})$.

\[
P(\text{observing } o \mid \text{ we see } s) = \sum_{p \in \text{Plant}} o(p) \times P(\text{observed plant is } p \mid \text{ we see } s)
\]

Thus we can also let $\text{Species} = \mathcal{D}(\text{Plant})$!
An example

type color = Red | Blue | White | Rose

type plant = { flower_color: color }
An example

type color = Red | Blue | White | Rose

type plant = { flower_color: color }

type observation = plant distribution

let whiteish : observation = fun p ->
    match p.flower_color with
    | White -> 0.8
    | Rose -> 0.2
    | _ -> 0.

let laurier_rose : species = fun p ->
    match p.flower_color with
    | Rose -> 0.8
    | White -> 0.2
    | _ -> 0.
An example

type color = Red | Blue | White | Rose

type plant = { flower_color: color }

type observation = plant distribution

let whiteish : observation = fun p ->
    match p.flower_color with
    | White -> 0.8
    | Rose -> 0.2
    | _ -> 0.

let laurier_rose : species = fun p ->
    match p.flower_color with
    | Rose -> 0.8
    | White -> 0.2 (* cultivars *)
    | _ -> 0.0

Where are we?

We have all the conceptual ingredients to make it work:

- The decision tree \textbf{algorithm}
- Species and observation with \textbf{uncertainty}
- How to make species and observation \textbf{interact} (score)
Where are we?

We have all the conceptual ingredients to make it work:

- The decision tree **algorithm**
- Species and observation with **uncertainty**
- How to make species and observation **interact** (score)

However:

- What is the real type for plant?
- Where do we get the data?
- If plant is big, isn't score intractable?
1. How to have expert botanists write the type plant?

2. How to describe the distribution probabilities for species?
   - User-friendliness: we need a lot of data, hence of a lot of people!
   - Link formal/informal: bibliographical info linked to data
The Flat Model – Plant

The naive approach:

\[ \text{Plant} = \prod_{i \in I} C_i \]

where:

- \( I \) is the set of **characters**
- Each \( C_i \) is a simple sum \( 1 + \ldots + 1 \)
The Flat Model – Plant

The naive approach:

\[
\text{Plant} = \prod_{i \in I} C_i
\]

where:

- \( I \) is the set of \textit{characters}
- Each \( C_i \) is a simple sum \( 1 + \ldots + 1 \)

Can be described by a textual format:

- \text{flower-color} = [ red blue white rose ... ];
- \text{flower-petal-number} = [ 1 2 3 ... ];
- \text{leaf-structure} = [ simple divided ];
Distributions over Plant are inconvenient. However, we have a correspondance:

\[ \mathcal{D}(S \times T) \leftrightarrow \mathcal{D}(S) \times \mathcal{D}(T) \]
Distributions over Plant are inconvenient. However, we have a correspondence:

\[ \mathcal{D}(S \times T) \Leftrightarrow \mathcal{D}(S) \times \mathcal{D}(T) \]

Thus we use

\[ \mathcal{D}({\text{Plant}}) \Leftrightarrow \prod_{i \in I} \mathcal{D}(C_i) \]
Distributions over Plant are inconvenient. However, we have a correspondance:

$$\mathcal{D}(S \times T) \iff \mathcal{D}(S) \times \mathcal{D}(T)$$

Thus we use

$$\mathcal{D}({\text{Plant}}) \iff \prod_{i \in I} \mathcal{D}(C_i)$$

# laurier-rose.species
flower-color = [ white: 0.8 rose: 0.2 ];
# we can omit the rest and use uniform distribution
The Flat Model – Species

Distributions over Plant are inconvenient. However, we have a correspondance:

\[ \mathcal{D}(S \times T) \Leftrightarrow \mathcal{D}(S) \times \mathcal{D}(T) \]

Thus we use

\[ \mathcal{D}(\text{Plant}) \Leftrightarrow \prod_{i \in I} \mathcal{D}(C_i) \]

# laurier-rose.species
flower-color = [ white: 0.8 rose: 0.2 ];
# we can omit the rest and use uniform distribution

With this representation, score can be computed in time \( O(C_1 + \ldots + C_n) \) instead of \( O(C_1 \times \ldots \times C_n) \)!
Limits of the flat model

1. Plant is too **rigid**: what about plants without flowers?

2. Species cannot represent all probability distributions.
   No **correlation** between trait distributions.

3. Compositional structure on species?
   Merge different descriptions ...
Going full algebraic type

What if we allow Plant to be an algebraic type?

```
plant := {
  leaf;
  flower;
}.
leaf := {
  position: [ base | stem {disposition} ];
  venation;
  attachment;
}.
flower := {
  inflorescence;
  sex: [ unisexual | hermaphrodisim ];
  color: [ red | blue | white | rose ];
}.
```

⇝ Similar to ontologies (e.g. RDFS).
What about Species?

We need to extend our **abstraction**:

\[ \mathcal{D}(S \times T) \leftrightarrow \mathcal{D}(S) \times D(T) \]

to:

\[ \mathcal{D}(S \oplus T) \leftrightarrow \]
What about Species?

We need to extend our \textbf{abstraction}:

\[
\mathcal{D}(S \times T) \leftrightarrow \mathcal{D}(S) \times D(T)
\]

to:

\[
\mathcal{D}(S \oplus T) \leftrightarrow [0, 1] \times \mathcal{D}(S) \times D(T)
\]

Example of a distribution:

\[
\begin{aligned}
\{ 
\text{leaf} &= \{ \text{position} = \text{stem} \}; \\
\text{flower} &= \{ \text{color} = [0.8: \text{rose} \mid 0.2: \text{white}] \} 
\}
\end{aligned}
\]
What about Species?

We need to extend our **abstraction**:

\[ \mathcal{D}(S \times T) \rightleftharpoons \mathcal{D}(S) \times \mathcal{D}(T) \]

to:

\[ \mathcal{D}(S \oplus T) \rightleftharpoons [0, 1] \times \mathcal{D}(S) \times \mathcal{D}(T) \]

Example of a distribution:

\{ leaf = { position = stem };  
   flower = { color = [ 0.8: rose | 0.2: white ] }  
\}

Abstract type interpretation:

\[ [S \times T] = [S] \times [T] \quad [S + T] = [0, 1] \times [S] \times [T]. \]
Biological species model of Linear Logic

We cannot still express **polymorphism**: 

\[
\text{simple basal leaves, compound stem leaves}
\]
We cannot still express **polymorphism**: 

\[
\text{simple basal leaves, compound stem leaves}
\]

\[
\implies \text{Can only say: simple or compound, basal or compound.}
\]
Biological species model of Linear Logic

We cannot still express **polymorphism**:

\[ \text{simple basal leaves, compound stem leaves} \]

\[ \rightsquigarrow \text{Can only say: simple or compound, basal or compound.} \]

We add an exponential \( !S \) to the type language:

\[ [!S] = \mathcal{D}(S) \]

We can thus write:

\[ [0.5 \{ \text{leaf} = \{ \text{simple} ; \text{basal} \} \}] \]
\[ | 0.5 \{ \text{leaf} = \{ \text{compound}; \text{stem} \} \} ] \]

In the plant description, we can add modalities:

\[ \text{plant} = \{ \text{leaf!}; \text{flower!}; \text{stem} \} \]
Conclusion

We have a working prototype for the flat model:

- An editor for entering species distribution
- Greedy algorithm implementation
Conclusion

We have a working prototype for the flat model:

- An editor for entering species distribution
- Greedy algorithm implementation

Inria exploratory project *Back to the trees*:

- Extend the model
- Work with local associations to build a database
- Implement non-greedy algorithms?