

Probabilistic reasoning with complex heterogeneous observations

David Poole

Department of Computer Science,
University of British Columbia

Work with: <http://minervaintelligence.com>, <https://treatment.com/>

February 5, 2021

“The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people.”

...

“In a universe with any regularities at all, decisions informed about the past are better than decisions made at random. That has always been true, and we would expect organisms, especially informavores such as humans, to have evolved acute intuitions about probability. The founders of probability, like the founders of logic, assumed they were just formalizing common sense.”

Steven Pinker, *How the Mind Works*, 1997, pp. 524, 343.

*“The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and **probabilistic reasoning about plants, animals, objects, and people.**”*

...

“In a universe with any regularities at all, decisions informed about the past are better than decisions made at random. That has always been true, and we would expect organisms, especially informavores such as humans, to have evolved acute intuitions about probability. The founders of probability, like the founders of logic, assumed they were just formalizing common sense.”

Steven Pinker, *How the Mind Works*, 1997, pp. 524, 343.

Outline

- Triples and Reification

Choosing Individuals and Relations in Logic

First-order logical languages allow many different ways of representing facts.

E.g., How to represent: “Pen #7 is red.”

Choosing Individuals and Relations in Logic

First-order logical languages allow many different ways of representing facts.

E.g., How to represent: “Pen #7 is red.”

- $red(pen_7)$. It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen_7 ?”

Choosing Individuals and Relations in Logic

First-order logical languages allow many different ways of representing facts.

E.g., How to represent: “Pen #7 is red.”

- $red(pen_7)$. It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen_7 ?”
- $color(pen_7, red)$. It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of pen_7 ?”
Can’t ask “What property of pen_7 has value red ?”

Choosing Individuals and Relations in Logic

First-order logical languages allow many different ways of representing facts.

E.g., How to represent: “Pen #7 is red.”

- $red(pen_7)$. It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen_7 ?”
- $color(pen_7, red)$. It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of pen_7 ?”
Can’t ask “What property of pen_7 has value red ?”
- $prop(pen_7, color, red)$. It’s easy to ask all these questions.

Choosing Individuals and Relations in Logic

First-order logical languages allow many different ways of representing facts.

E.g., How to represent: “Pen #7 is red.”

- $red(pen_7)$. It’s easy to ask “What’s red?”
Can’t ask “what is the color of pen_7 ?”
- $color(pen_7, red)$. It’s easy to ask “What’s red?”
It’s easy to ask “What is the color of pen_7 ?”
Can’t ask “What property of pen_7 has value red ?”
- $prop(pen_7, color, red)$. It’s easy to ask all these questions.

With a single relation it can be implicit \rightarrow triples:

$\langle pen_7, color, red \rangle$.

Universality of *prop*

To represent “a is a parcel”

Universality of *prop*

To represent “a is a parcel”

- $prop(a, type, parcel)$, where *type* is a special property and *parcel* is a class.
- $prop(a, parcel, true)$, where *parcel* is a Boolean property (characteristic function of the class *parcel*).

Triples

- To represent *tutorial("StarAI", nips2017, 1045, hallC)*. "the Star AI tutorial at NIPS 2017 is scheduled at 10:45 in Hall C."

Triples

- To represent *tutorial("StarAI", nips2017, 1045, hallC)*. “the Star AI tutorial at NIPS 2017 is scheduled at 10:45 in Hall C.”
- Let *t123* name the offering of the tutorial:

prop(t123, type, tutorial).

prop(t123, title, "StarAI").

prop(t123, event, nips2017).

prop(t123, time, 1045).

prop(t123, room, hallC).

Triples

- To represent *tutorial*("StarAI", *nips2017*, 1045, *hallC*). "the Star AI tutorial at NIPS 2017 is scheduled at 10:45 in Hall C."
- Let *t123* name the offering of the tutorial:

prop(t123, type, tutorial).

prop(t123, title, "StarAI").

prop(t123, event, nips2017).

prop(t123, time, 1045).

prop(t123, room, hallC).

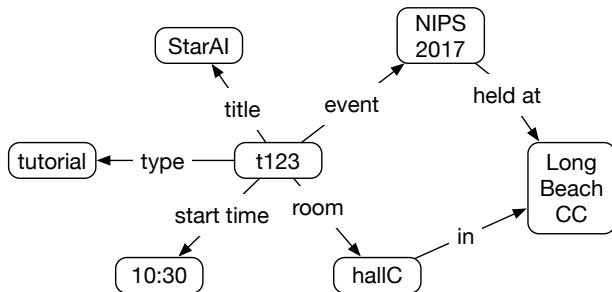
- We have **reified** the booking.
- Reify means: to make into an individual.

Triples

- To represent *tutorial*("StarAI", *nips2017*, 1045, *hallC*). "the Star AI tutorial at NIPS 2017 is scheduled at 10:45 in Hall C."
- Let *t123* name the offering of the tutorial:
 - prop(t123, type, tutorial)*.
 - prop(t123, title, "StarAI")*.
 - prop(t123, event, nips2017)*.
 - prop(t123, time, 1045)*.
 - prop(t123, room, hallC)*.
- We have **reified** the booking.
- Reify means: to make into an individual.
- How can we add extra arguments (e.g., presenters, chair)?

Triples and Knowledge Graphs

- Subject–verb–object
Individual–property–value
triples can be depicted as edges in graphs



- A modeller or learner needs to invent (reified) objects.

Triples are universal representations of relations

All relations can be represented in terms of **triples**:

	...	P_j	...
r_i
	...	v_{ij}	...

can be represented as

Triples are universal representations of relations

All relations can be represented in terms of **triples**:

	...	P_j	...
r_i
	...	v_{ij}	...

can be represented as the triple $\langle r_i, P_j, v_{ij} \rangle$.

- r_i is either a primary key or a **reified** individual.

Triples are universal representations of relations

All relations can be represented in terms of **triples**:

	...	P_j	...
r_i
	...	v_{ij}	...

can be represented as the triple $\langle r_i, P_j, v_{ij} \rangle$.

- r_i is either a primary key or a **reified** individual.
- **Examples of reified individuals**: a booking, a marriage, a talk, a lab report, an event, a party, a meeting, a drink

Triples are universal representations of relations

All relations can be represented in terms of **triples**:

	...	P_j	...
r_i
	...	v_{ij}	...

can be represented as the triple $\langle r_i, P_j, v_{ij} \rangle$.

- r_i is either a primary key or a **reified** individual.
- **Examples of reified individuals**: a booking, a marriage, a talk, a lab report, an event, a party, a meeting, a drink
- Challenge for learning: each reified individual has limited data to learn from; (at most) one value for each property.

Triples are universal representations of relations

All relations can be represented in terms of **triples**:

	...	P_j	...
r_i
	...	v_{ij}	...

can be represented as the triple $\langle r_i, P_j, v_{ij} \rangle$.

- r_i is either a primary key or a **reified** individual.
- **Examples of reified individuals**: a booking, a marriage, a talk, a lab report, an event, a party, a meeting, a drink
- Challenge for learning: each reified individual has limited data to learn from; (at most) one value for each property.

$\text{prop}(\text{Individual}, \text{Property}, \text{Value})$ is the only relation needed:
 $\langle \text{Individual}, \text{Property}, \text{Value} \rangle$ triples, Semantic network, entity relationship model, knowledge graphs, ...

What is now required is to give the greatest possible development to mathematical logic, to allow to the full the importance of relations, and then to found upon this secure basis a new philosophical logic, which may hope to borrow some of the exactitude and certainty of its mathematical foundation. If this can be successfully accomplished, there is every reason to hope that the near future will be as great an epoch in pure philosophy as the immediate past has been in the principles of mathematics. Great triumphs inspire great hopes; and pure thought may achieve, within our generation, such results as will place our time, in this respect, on a level with the greatest age of Greece.

– Bertrand Russell 1917