Practical AI: reasoning

Stanislav Protasov for
Harbour.Space

Refresh

What is function?

What are the ways we can implement intelligent function?

Agenda

- Data, Information, Knowledge, Wisdom
- Declarative and imperative languages
- Decision trees
- Math software
- Production rules
- Ontologies, Bayesian networks

Knowledge, information, data

Knowledge - knowing something that allows us to make decisions [and passes empirical check]. Knowledge can be encoded with *world state*, *triggers* and *actions*.

Knowledge of a robot example:

1) **Fact**: This human is dangerous for me
2) **Rule**: If human is dangerous - destroy him
3) **Algorithms**: To destroy a human, catch a human ...

Utility scale:

Data -> Information -> Knowledge -> Wisdom

Knowledge

Educated guess: to be **transferred** and **stored**, knowledge is represented using **language**

- Natural language
- Formal language
- Other languages

Declarative vs imperative programming languages

Software written in Java/C++/python... can be treated as *knowledge*, as soon as it encodes **rules**, **algorithms** and sometimes facts. These languages are called **imperative**, because ordered operators *command*, how to change program state.

Declarative languages do not focus on algorithms, but define (declare) **facts** and **rules**. Algorithms are usually implicit.

Declarative languages:

- Domain specific: markup HTML, query SQL and xquery, math
- Functional (Haskell)
- Logic (prolog, CLIPS, ...)
- ...

Decision Trees

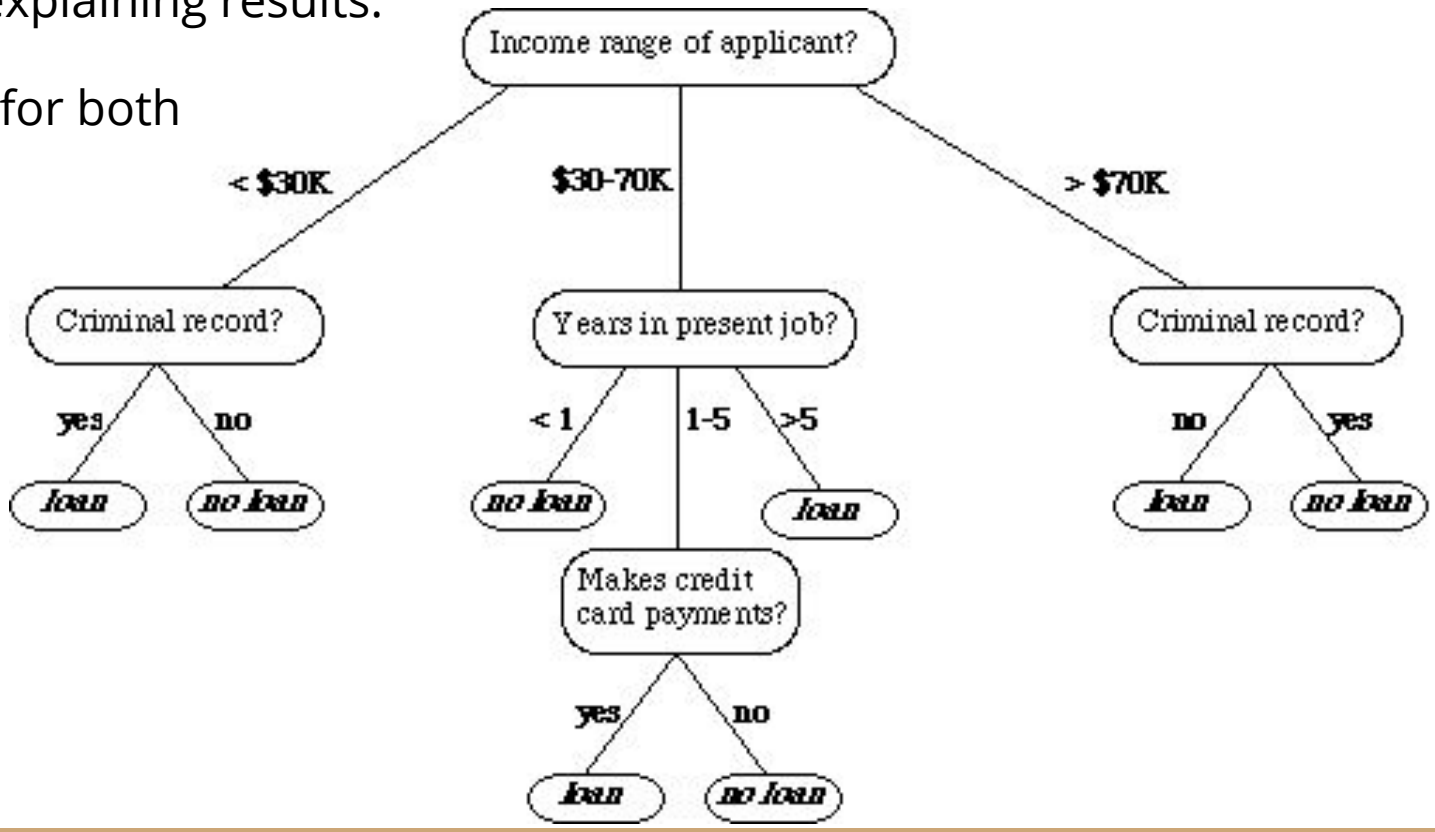
Initially, decision trees were created manually to encode knowledge.

Later they were implemented as software in expert systems.

They are cool in explaining results.

DTs can we used for both classification and regression.

Today DT are used in‘ machine learning.



Lab #1

Write a function, that implement image on the previous slide

- Think how to encode tree and nodes
- Create decision tree for this task
- Store that tree in a file in human-readable format
- Write the code that uses the tree

Math systems

Math is the most universal language.
Math has a lot of universal formal rules and algorithms.
Major math tools:

- **Automated theorem proving**
  - Proving, prove check
  - satisfiability
  - code verification, ...
- **Computer algebra systems:**
  - Matlab/Octave,
  - Maple,
  - Wolfram Mathematica & Alpha
  - Maxima,
  - ...

```
(%i2)  g(x) := x^2;
```

$$g(x) := x^2$$

```
(%i3)  solve(g(x)=4);
```

$$[x=-2,x=2]$$

```
(%i6)  simplify(x^2 - x^2 + x / 2 + 0.5 * x);
```

$$\text{simplify}(1.0\,x)$$

```
(%i9)  f(x) := integrate(x^2, x);
```

$$f(x) := \int x^2\,dx$$

```
(%i11)  f(x=3);
```

$$\int x^2\,dx=3 = \%c1 + \int 9\,dx=3$$

```
(%i15)  solve(f(x) = 9, x);
```

$$[x=\frac{3^{3/2}\,\%i-3}{2},x=-\frac{3^{3/2}\,\%i+3}{2},x=3]$$

NPV example

$$NPV = -C_0 + \frac{C_1}{1+r} + \frac{C_2}{(1+r)^2} + ... + \frac{C_T}{(1+r)^T}$$

$-C_0 = Initial\ Investment$
$C = Cash\ Flow$
$r = Discount\ Rate$
$T = Time$

Net Present Value (NPV) is a formula used to determine the present value of an investment by the discounted sum of all cash flows received from the project.

Lab #2

Now you know NPV formula. Write a function (using provided template), that will evaluate discount rate if given NPV > 0 and cash flow.

Production rules

Most **expert systems** are represented as **production systems**.

Cool thing about productions systems — **backtracking**.

Productions consist of two parts:

- a **sensory precondition** (or "IF" statement) and
- an **action** (or "THEN").

Most expert systems apply forward chaining to dig new facts, which is just repeated application of
modus ponens $((P \rightarrow Q) \land P) \rightarrow Q$

The most widely used tool for expert systems is CLIPS.

CLIPS example

```
(deftemplate human
    (slot name) (slot age) (slot gender (default male)))
(defrule gotoarmy
    (human (name ?x) (age 18) (gender male))
    (season autumn|spring)
    =>
    (assert (serve ?x)))
; assertions
(assert (human (name "Alex") (age 18)))
(assert (season spring))
```

… facts

```
f-1  (human (name "Alex") (age 18) (gender male))

f-2  (season spring)

f-3  (serve "Alex")
```

Lab #3

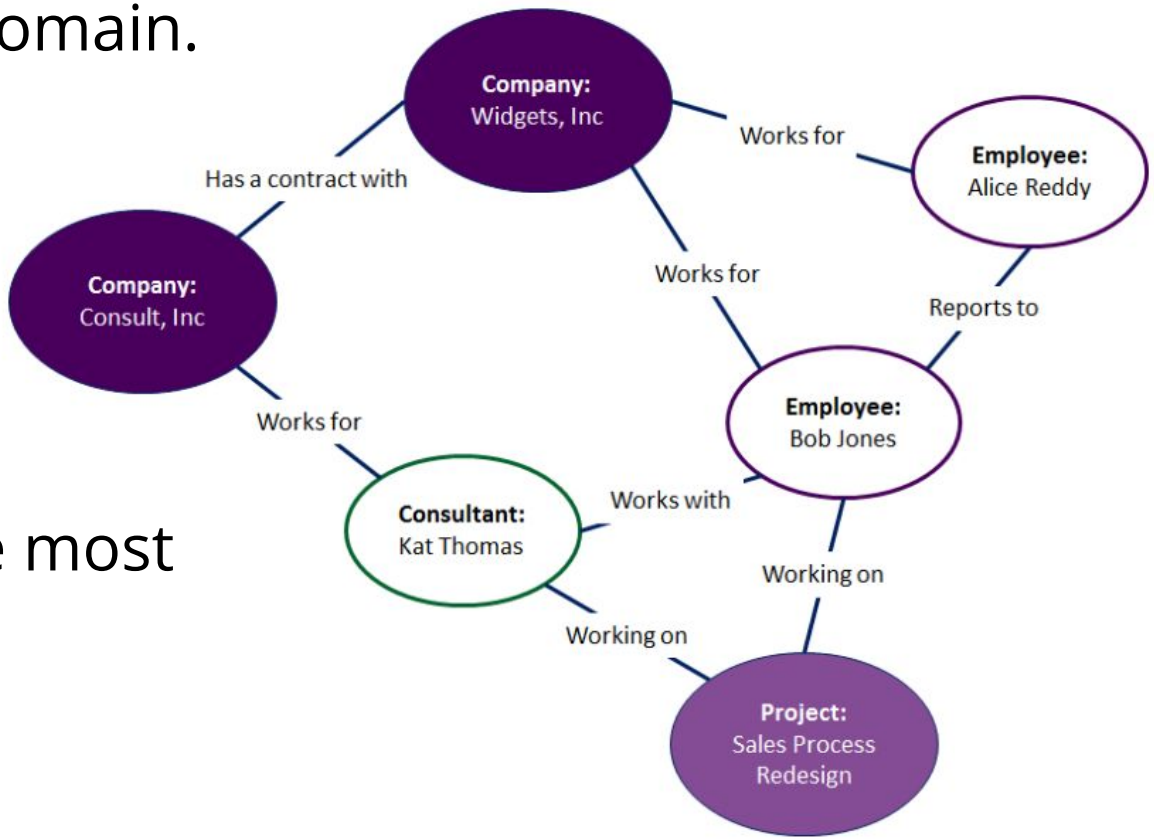
Find shortest path between stations given station connections

Please refer to a template

Use any IDE: standalone or online e.g. this or this.

Ontology

Ontology is a formal naming and definition of the *types*, *properties*, and *interrelationships* of the entities that really exist in a particular domain.



OWL and **Cyc** are the most widely used systems

Frames by Marvin Minsky

Frames were proposed in 1974 to represent knowledge. They have very similar to ontologies idea, but also provide a tool for behavior (called *procedures and scenarios*).

Frame model is a conceptual basis for *OOP*.

==========

Expert systems, ontologies, and other domain-specific tools should be entered by an *expert*, but designed for *engineers*

Bayesian network

Conditional probabilities can be considered as an extension of production rules:

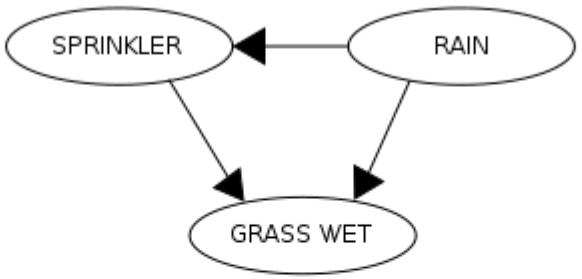
IF (A) THEN (B) => P(B|A) = 1 => P(B|A) = 0.9

Bayesian networks used to encode production can solve following problems:

- Evidence probability
- Reason probability
- Most probable explanation
- Prediction
- ...

$$\mathrm{P}(G, S, R) = \mathrm{P}(G|S, R)\mathrm{P}(S|R)\mathrm{P}(R)$$

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



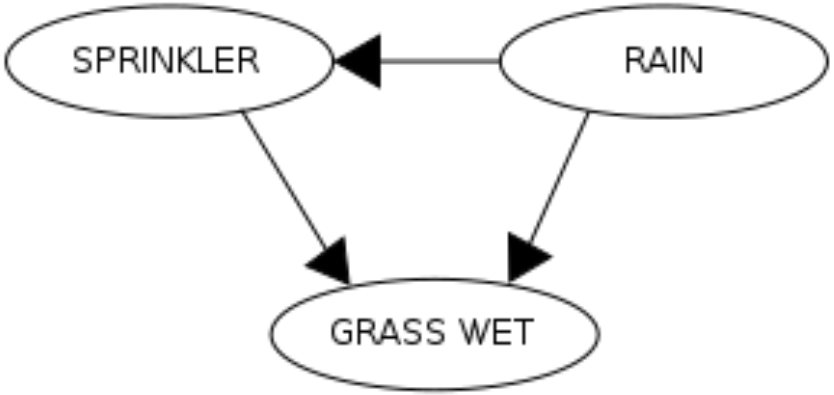
RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

One of widely used tools is **Hugin Expert**

Whiteboard time

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

I see **wet grass**. What is the **probability of rain outside**?

Hometask #1

Here is the game in CLIPS https://md5crypt.github.io/clipsgame/. Solve at least 2 levels of the game. Submit your solution (full script with your additional facts) with comments to github.

Hometask #2. Forward reasoning

Given a Bayesian graph like in example implement forward reasoning.

1) Graph nodes and edges are defined with conditional probabilities
2) If given initial probability of start node(s) (i.e. P(rain) = 0.2) compute full probabilities of all other nodes.