Artificial Intelligence & Machine Learning and Pattern Recognition ——Logic

Yanghui Rao
Assistant Prof., Ph.D
School of Mobile Information Engineering,
Sun Yat-sen University
raoyangh@mail.sysu.edu.cn

Logic (逻辑)

- Logic is a great knowledge representation language for many AI problems
- **Propositional logic** (命题逻辑) is the foundation and fine for some AI problems
- **First order Predicate logic** (一阶谓词逻辑) is much more expressive and more commonly used in AI
- Many variations: higher order predicate logic, three-valued logic, probabilistic logics, etc.

PL (命题逻辑)

- **Logical constants**: true, false
- Propositional **symbols**: P, Q,... (**atomic sentences**)
- Wrapping **parentheses**: (…)
- Sentences are combined by **connectives**:

$\wedge$	and	[conjunction]
$\vee$	or	[disjunction]
$\Rightarrow$	implies (蕴含)	[implication / conditional]
$\Leftrightarrow$	is equivalent (等价)	[equivalence]
$\neg$	not	[negation]

- $P \wedge Q$, $\neg P \wedge Q$, $\neg P \vee Q$, $((P) \vee Q)$, etc.

PL (命题逻辑)

- Simple language for showing key ideas and definitions
- User defines **semantics** of each propositional symbol:
  - P means "It is hot", Q means "It is humid", etc.
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg$S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then ($S \vee T$), ($S \wedge T$), ($S \Rightarrow T$), and ($S \Leftrightarrow T$) are sentences
  - If expressions are parenthesized, the term in the parentheses is evaluated first. Otherwise, the priorities are: $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

Examples of PL Sentences

- Q
  "It is humid."
- $Q \Rightarrow P$
  "If it is humid, then it is hot"
- $(P \wedge Q) \Rightarrow R$
  "If it is hot and humid, then it is raining"
- We're free to choose better symbols, e.g.,
  Ho = "It is hot"
  Hu = "It is humid"
  R = "It is raining"

Truth Tables

- Truth tables are used to define logical connectives and to determine when a complex sentence is true given the values of the symbols in it
- Note that $\Rightarrow$ is a logical connective, so $P \Rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false

Truth tables for the five logical connectives

P	**Q**	$\neg$ **P**	**P** $\wedge$ **Q**	**P** $\vee$ **Q**	**P** $\Rightarrow$ **Q**	**P** $\Leftrightarrow$ **Q**
False	False					
False	True					
True	False					
True	True					

Truth Tables

- Truth tables are used to define logical connectives and to determine when a complex sentence is true given the values of the symbols in it
- Note that $\Rightarrow$ is a logical connective, so $P \Rightarrow Q$ is a logical sentence and has a truth value, i.e., is either true or false

Truth tables for the five logical connectives

P	**Q**	**$\neg$ P**	**P $\wedge$ Q**	**P $\vee$ Q**	**P $\Rightarrow$ Q**	**P $\Leftrightarrow$ Q**
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Truth Tables

Example of a truth table used for a complex sentence

P	**Q**	**$(P \lor Q) \land (\neg Q)$**	**$((P \lor Q) \land (\neg Q)) \Rightarrow P$**
False	False		
False	True		
True	False		
True	True		

Truth Tables

Example of a truth table used for a complex sentence

P	**Q**	**$(P \lor Q) \land (\neg Q)$**	**$((P \lor Q) \land (\neg Q)) \Rightarrow P$**
False	False	False	True
False	True	False	True
True	False	True	True
True	True	False	True

Knowledge Base (KB)

- The meaning or **semantics** of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its **truth value** (True or False)
- A **model** for a knowledge base (**KB**) is a *possible world* – an assignment of truth values to propositional symbols that makes each sentence in the KB True

Model for a KB

- Let the KB be [$P \wedge Q \Rightarrow R$, $Q \Rightarrow P$]
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables
  - **FFF:**
  - **FFT:**
  - **FTF:**
  - **FTT:**
  - **TFF:**
  - **TFT:**
  - **TTF:**
  - **TTT:**

P: it's hot
Q: it's humid
R: it's raining

Model for a KB

- Let the KB be [$P \wedge Q \Rightarrow R$, $Q \Rightarrow P$]
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables
  - **FFF: OK**
  - **FFT: OK**
  - FTF: NO
  - FTT: NO
  - **TFF: OK**
  - **TFT: OK**
  - TTF: NO
  - **TTT: OK**
- If KB is [$P \wedge Q \Rightarrow R$, $Q \Rightarrow P$, Q], then the answer is ?

P: it's hot
Q: it's humid
R: it's raining

Model for a KB

- Let the KB be [$P \wedge Q \Rightarrow R$, $Q \Rightarrow P$]
- What are the possible models? Consider all possible assignments of T|F to P, Q and R and check truth tables
  - **FFF: OK**
  - **FFT: OK**
  - FTF: NO
  - FTT: NO
  - **TFF: OK**
  - **TFT: OK**
  - TTF: NO
  - **TTT: OK**
- If KB is [$P \wedge Q \Rightarrow R$, $Q \Rightarrow P$, Q], then the answer is **TTT**

P: it's hot
Q: it's humid
R: it's raining

Pros and Cons of PL

- + Meaning of propositional logic is context independent: (unlike natural language, where the meaning depends on the context)
- - Propositional logic has limited expressive power: (unlike natural language)
  - “ Robot A is to the right of robot B”
  - Robot_3_is_to_the_right_of_robot_9 ⇔

    Robot_3_is_situated_at_*xy*_postition_(35, 79)

    ∧ Robot_9_is_situated_at_*xy*_postition_(10, 93)

    ∨ …

First-order Predicate Logic

- **Objects** (个体词): represent a specific object by a, b, …
- **Predicate** (谓词): represent the attribute of objects by $A(\ldots)$, $B(\ldots)$, $\ldots Z(\ldots)$
  - **Relations** (关系), e.g., bigger than, inside, part of, …
  - **Functions** (性质), e.g., red, round, …
- **Quantifier** (量词)
  - **universal quantifier:** $\forall$
  - **existential quantifier:** $\exists$

$\forall x$ Frog $(x) \Rightarrow$ Green (x):

$\neg\, \forall x$ Likes $(x$, cat):

$\neg\, \exists x$ Likes $(x$, cat):

First-order Predicate Logic

- **Objects** (个体词): represent a specific object by a, b, …
- **Predicate** (谓词): represent the attribute of objects by A (…), B (…), …Z (…)
  - **Relations** (关系), e.g., bigger than, inside, part of, …
  - **Functions** (性质), e.g., red, round, …
- **Quantifier** (量词)
  - **universal quantifier:** $\forall$
  - **existential quantifier:** $\exists$

$\forall x$ Frog (x) $\Rightarrow$ Green (x): All frogs are green
$\neg \forall x$ Likes (x, cat): Not everyone likes cat
$\neg \exists x$ Likes (x, cat): No one likes cat

First-order Predicate Logic

- ✓ " Robot A is to the right of robot B"
- ✓ $\forall u\ \forall v$ is_further_right$(u, v) \Leftrightarrow$

  $\exists x_u\ \exists y_u\ \exists x_v\ \exists y_v$ Position$(u, x_u, y_u) \wedge$ Position(v, x_v, y_v)

  $\wedge$ Larger(x_u, x_v)

- Typically, $\Rightarrow$ is the main connective with $\forall$;

  $\wedge$ is the main connective with $\exists$
  - $\forall x$ At$(x$, SMIE$) \Rightarrow$ Smart (x)
  - $\exists x$ At$(x$, SMIE$) \wedge$ Smart (x)
- **Morgan's law**
  - $\forall x\ L \equiv \neg\ \exists x\ \neg L$
  - $\neg(\forall x\ L) \equiv \exists x\ \neg L$

First-order Predicate Logic

- ✓ " Robot A is to the right of robot B"
- ✓ $\forall u\ \forall v$ is_further_right$(u, v) \Leftrightarrow$

  $\exists x_u\ \exists y_u\ \exists x_v\ \exists y_v\ \mathrm{Position}(u, x_u, y_u) \wedge \mathrm{Position}(v, x_v, y_v)$

  $\wedge\ \mathrm{Larger}(x_u, x_v)$

- Typically, $\Rightarrow$ is the main connective with $\forall$; $\wedge$ is the main connective with $\exists$
  - $\forall x\ \mathrm{At}(x, \mathrm{SMIE}) \Rightarrow \mathrm{Smart}\ (x)$
  - $\exists x\ \mathrm{At}(x, \mathrm{SMIE}) \wedge \mathrm{Smart}\ (x)$
- **Morgan's law**
  - $\forall x\ L \equiv \neg\ \exists x\ \neg L$
  - $\neg(\forall x\ L) \equiv \exists x\ \neg L$

"Not everyone likes cat"

$\neg(\forall x,\ \mathrm{Likes}(x, \mathrm{cat}))$

$\exists x, \neg\mathrm{Likes}(x, \mathrm{cat})$

Quantifier Scope

- If a quantifier Q is followed by (, then the scope of Q is to the matched)
  - $\forall x\,(F(x) \Leftrightarrow F(h))$

Quantifier Scope

- If a quantifier Q is followed by (, then the scope of Q is to the matched)
  - $\forall x\, (F(x) \Leftrightarrow F(h))$
- If a quantifier Q is not followed by (or another quantifier, then the scope of Q is to the first connective
  - $\forall x\, F(x) \Leftrightarrow F(h)$

Quantifier Scope

- If a quantifier Q is followed by (, then the scope of Q is to the matched)
  - $\forall x\ (F(x) \Leftrightarrow F(h))$
- If a quantifier Q is not followed by (or another quantifier, then the scope of Q is to the first connective
  - $\forall x\ F(x) \Leftrightarrow F(h)$
- If a quantifier $Q1$ is followed by another quantifier $Q2$, then the scope of $Q1$ is to the scope of $Q2$
  - $\forall x\ \exists y\ R(x, y)$

Quantifier Scope

- If a quantifier Q is followed by (, then the scope of Q is to the matched)
  - $\forall x\ (F(x) \Leftrightarrow F(h))$
- If a quantifier Q is not followed by (or another quantifier, then the scope of Q is to the first connective
  - $\forall x\ F(x) \Leftrightarrow F(h)$
- If a quantifier $Q1$ is followed by another quantifier $Q2$, then the scope of $Q1$ is to the scope of $Q2$
  - $\forall x\ \exists y\ R(x, y)$
- F: … can fly
- h: human being

$$\forall x\ (F(x) \Leftrightarrow F(h)) \overset{?}{\Leftrightarrow} \forall x\ F(x) \Leftrightarrow F(h)$$

Quantifier Scope

- If a quantifier Q is followed by (, then the scope of Q is to the matched)
  - $\underline{\forall x\, (F(x) \Leftrightarrow F(h))}$
- If a quantifier Q is not followed by (or another quantifier, then the scope of Q is to the first connective
  - $\underline{\forall x\, F(x)} \Leftrightarrow F(h)$
- If a quantifier $Q1$ is followed by another quantifier $Q2$, then the scope of $Q1$ is to the scope of $Q2$
  - $\underline{\forall x\, \underline{\exists y\, R(x, y)}}$
- F: … can fly
- h: human being

False: $\forall x\, (F(x) \Leftrightarrow F(h))$ $\not\Leftrightarrow$ True: $\forall x\, F(x) \Leftrightarrow F(h)$

Interacting with KBs

- Tell the system assertions
  - Facts :
    - Tell (KB, Bird(eagle))
    - Tell (KB, Penguin企鹅(Tweety))

  - Rules:
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow$ Bird(x))
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow \neg$ Fly(x))
    - Tell (KB, $\forall x$ Bird(x) $\Rightarrow$ Fly(x))
- Ask questions
  - Ask (KB, Bird(eagle))
  - Ask (KB, Fly(eagle))
  - Ask (KB, Fly(Tweety))

Interacting with KBs

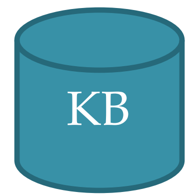
- Tell the system assertions
  - Facts :
    - Tell (KB, Bird(eagle))
    - Tell (KB, Penguin(Tweety))
  - Rules:
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow$ Bird(x))
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow \neg$ Fly(x))
    - Tell (KB, $\boxed{\forall x \text{ Bird}(x) \wedge \neg \text{ Penguin}(x) \Rightarrow \text{Fly}(x))}$
- Ask questions
  - Ask (KB, Bird(eagle))
  - Ask (KB, Fly(eagle))
  - Ask (KB, Fly(Tweety))

Interacting with KBs

- Tell the system assertions
  - Facts :
    - Tell (KB, Bird(eagle))
    - Tell (KB, Penguin(Tweety))
    - Tell (KB, Raven乌鸦(abraxas)
  - Rules:
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow$ Bird(x))
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow \neg$ Fly(x))
    - Tell (KB, $\forall x$ Bird(x) $\wedge \neg$ Penguin(x) $\Rightarrow$ Fly(x))
    - Tell (KB, $\forall x$ Raven(x) $\Rightarrow$ Bird(x))
- Ask questions
  - Ask (KB, Bird(eagle))
  - Ask (KB, Fly(eagle))
  - Ask (KB, Fly(Tweety))
  - Ask (KB, Fly(abraxas)?

Interacting with KBs

- Tell the system assertions
  - Facts :
    - Tell (KB, Bird(eagle))
    - Tell (KB, Penguin(Tweety))
    - Tell (KB, Raven(abraxas))
  - Rules:
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow$ Bird(x))
    - Tell (KB, $\forall x$ Penguin(x) $\Rightarrow \neg$ Fly(x))
    - Tell (KB, $\forall x$ Bird(x) $\wedge \neg$ Penguin(x) $\Rightarrow$ Fly(x))
    - Tell (KB, $\forall x$ Raven(x) $\Rightarrow$ Bird(x))



- Ask questions
  - Ask (KB, Bird(eagle))
  - Ask (KB, Fly(eagle))
  - Ask (KB, Fly(Tweety))
  - Ask (KB, Fly(abraxas)?

Tell (KB, $\forall x$ Raven(x) $\Rightarrow \neg$ Penguin(x))

For the construction of a knowledge base with all 9,800 or so types of birds worldwide, it must therefore be specified for every type of bird (except for penguins) that it is not a member of penguins!