IS/SE/SS ZC444: Artificial Intelligence

12 Logic, and ML

Dr. Kamlesh Tiwari
Associate Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Oct 29, 2023 ONLINE WILP @ BITS-Pilani [July-Dec 2023]

http://ktiwari.in/ai

Recap: First Order Logic (Predicate Logic)

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i \qquad \frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

$$\frac{\phi}{\neg\neg\phi} \neg\neg i \qquad \frac{\neg\neg\phi}{\phi} \neg\neg e$$

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$$

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \rightarrow \psi} \rightarrow i$$

Backus-Naur Form (BNF)

There is a rule (syntax) to form sentences

$$\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \mid ComplexSentence \\
AtomicSentence & \rightarrow & True \mid False \mid P \mid Q \mid R \mid \ldots \\
ComplexSentence & \rightarrow & (\ Sentence\ ) \mid [\ Sentence\ ] \\
 & | & \neg\ Sentence \\
 & | & Sentence \wedge Sentence \\
 & | & Sentence \vee Sentence \\
 & | & Sentence \ \Rightarrow\ Sentence \\
 & | & Sentence \ \Leftrightarrow\ Sentence \\
\end{array}$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Rules for Disjunction

$$\frac{\phi}{\phi \vee \psi} \vee i_1 \qquad \frac{\psi}{\phi \vee \psi} \vee i_2 \qquad \frac{\phi \vee \psi \quad \boxed{\begin{array}{c}\phi \\ \vdots \\ x\end{array}} \quad \boxed{\begin{array}{c}\psi \\ \vdots \\ x\end{array}}}{x} \vee e$$

Show $p \vee q \vdash q \vee p$

1.	$p \vee q$	Premise
2.	p	Assumption
3.	$q \vee p$	$\vee i_2$
4.	q	Assumption
5.	$q \vee p$	$\vee i_1$
6.	$q \vee p$	$\vee e$ 1, 3, 5

- Show $q \rightarrow r \vdash p \vee q \rightarrow p \vee r$

Rules for Negation

$$\frac{\perp}{\phi} \perp e \qquad \frac{\phi \quad \neg\phi}{\perp} \neg e \qquad \frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \perp\end{array}}}{\neg\phi} \neg i$$

Show $\neg p \vee q \vdash p \rightarrow q$

1.	$\neg p \vee q$			
2.	$\neg p$	Premise	q	Premise
3.	p	Assumption	p	Assumption
4.	$\perp$	$\neg e$ 3,2	q	Copy 2
5.	q	$\perp e$ 4	$p \rightarrow q$	$\rightarrow i$ 3,4
6.	$p \rightarrow q$	$\rightarrow i$ 3,5		
7.	$p \rightarrow q$			$\vee e$ 1,2-6

- Show $p \rightarrow q, p \rightarrow \neg q \vdash \neg p$

Logical Equivalences

$(\alpha \wedge \beta)$	=	$(\beta \wedge \alpha)$	Commutativity of $\wedge$
$(\alpha \vee \beta)$	=	$(\beta \vee \alpha)$	Commutativity of $\vee$
$(\alpha \wedge \beta) \wedge \gamma$	=	$\alpha \wedge (\beta \wedge \gamma)$	Associativity of $\wedge$
$(\alpha \vee \beta) \vee \gamma$	=	$\alpha \vee (\beta \vee \gamma)$	Associativity of $\vee$
$\neg\neg\alpha$	=	α	Double negation elimination
$\alpha \rightarrow \beta$	=	$\neg\beta \rightarrow \neg\alpha$	Contraposition
$\alpha \rightarrow \beta$	=	$\neg\alpha \vee \beta$	Implication Elimination
$\alpha \leftrightarrow \beta$	=	$(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$	Biconditional Elimination
$\neg(\alpha \wedge \beta)$	=	$\neg\alpha \vee \neg\beta$	De Morgen
$\neg(\alpha \vee \beta)$	=	$\neg\alpha \wedge \neg\beta$	De Morgen
$\alpha \wedge (\beta \vee \gamma)$	=	$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$	Distribution of $\wedge$ on $\vee$
$\alpha \vee (\beta \wedge \gamma)$	=	$(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$	Distribution of $\vee$ on $\wedge$

Soundness and Completeness

- Soundness: doing right
- Completeness: full coverage

There are 10 defective bulbs in a box of 25.

- Mr. A gives me 10 bulbs none of them is defective
- Mr. B gives me 20 bulbs; 5 of them is defective

A is sound
B is complete

Evaluate a legal system "guilty until proven innocent" and "innocent until proven guilty"

What we want? both.

CNF, IMPL_FREE and NNF

Conjunctive normal form[1], implication free[2] and negative normal form[3]

Find CNF(NNF(IMPL_FREE(A)))

Where $A = \neg p \wedge q \rightarrow p \wedge (r \rightarrow q)$

$$\neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee q))$$
$$(p \vee \neg q) \vee (p \wedge (\neg r \vee q))$$
$$(p \vee \neg q \vee p) \vee (p \wedge \neg q \wedge \neg r \vee q)$$

[1] everything is conjunctions of disjunction
[2] no $\rightarrow$
[3] no double negation

Horn Clause

Formula that can be generated by H

$$\begin{aligned} P &::= \bot \mid \top \mid p \mid q \mid r \mid \ldots \\ A &::= P \mid P \wedge A \\ C &::= A \rightarrow P \\ H &::= C \mid C \wedge H \end{aligned} \tag{1}$$

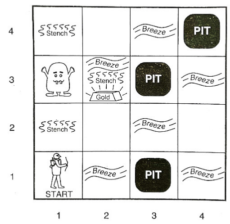
Satisfiability

1. It marks $\top$ if it occurs in that list.
2. If there is a conjunct $P_1 \wedge P_2 \wedge \cdots \wedge P_{k_i} \rightarrow P'$ of ϕ such that all P_j with $1 \leq j \leq k_i$ are marked, mark P' as well and go to 2. Otherwise (= there is no conjunct $P_1 \wedge P_2 \wedge \cdots \wedge P_{k_i} \rightarrow P'$ such that all P_j are marked) go to 3.
3. If $\bot$ is marked, print out 'The Horn formula ϕ is unsatisfiable.' and stop. Otherwise, go to 4.
4. Print out 'The Horn formula ϕ is satisfiable.' and stop.

(a) $(p \wedge q \wedge w \rightarrow \bot) \wedge (t \rightarrow \bot) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (u \rightarrow s) \wedge (\top \rightarrow u)$
(b) $(p \wedge q \wedge w \rightarrow \bot) \wedge (t \rightarrow \bot) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (r \wedge u \rightarrow w) \wedge (u \rightarrow s) \wedge (\top \rightarrow u)$
(c) $(p \wedge q \wedge s \rightarrow p) \wedge (q \wedge r \rightarrow p) \wedge (p \wedge s \rightarrow s)$
(d) $(p \wedge q \wedge s \rightarrow \bot) \wedge (q \wedge r \rightarrow p) \wedge (\top \rightarrow s)$
(e) $(p_5 \rightarrow p_{11}) \wedge (p_2 \wedge p_3 \wedge p_5 \rightarrow p_{13}) \wedge (\top \rightarrow p_5) \wedge (p_5 \wedge p_{11} \rightarrow \bot)$
(f) $(\top \rightarrow q) \wedge (\top \rightarrow s) \wedge (w \rightarrow \bot) \wedge (p \wedge q \wedge s \rightarrow \bot) \wedge (v \rightarrow s) \wedge (\top \rightarrow r) \wedge (r \rightarrow p)$

Recall Wumpus World

- **Performance** gold +100, death -100, step -1, arrow -10
- **Environment** smell around wumpus, breeze around pit
- **Actuator** turn left/right, forward, grab, release, shoot
- **Sensor** breeze, glitter, smell, bump, scream



Single Agent, Deterministic, Static, Discrete, !Observable & !Episodic

- $P_{x,y}$ if there is a pit in $[x, y]$
- $W_{x,y}$ if wumpus is in $[x, y]$
- $B_{x,y}$ if breeze is in $[x, y]$
- $S_{x,y}$ if stench is in $[x, y]$

We know R_1: $\neg P_{1,1}$, R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$, R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$, R_4: $\neg B_{1,1}$, R_5: $B_{2,1}$

Model Checking for Inference

- Seven symbols $P_{1,1}$, $B_{1,1}$, $P_{1,2}$, $P_{2,1}$, $B_{2,1}$, $P_{2,2}$, $P_{3,1}$ have $2^7 = 128$ models. In three of these knowledge base is true.

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

In all those three $\neg P_{1,2}$ is true, hence there is no pit in [1,2].
On the other hand $P_{2,2}$ is true on two and false in one so it is not confirmed whether there is pit in [2,2] or not.

Validity and Satisfiability

- **Validity**: sentence is true in all models (tautologies)

$$A \vee \neg A$$
$$A \vee B \rightarrow A \vee B$$

- **Satisfiability**: sentence is true in some models

$$A \vee \neg B$$
$$A \rightarrow B$$

Determine whether following sentence is valid or satisfiable

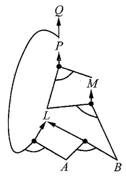
$$((A \wedge B) \rightarrow C) \leftrightarrow (A \rightarrow (B \rightarrow C))$$

Forward Chaining

Determines if a single proposition symbol q is entailed by the knowledge? (data driven reasoning)

- It begins from known facts and adds conclusions of the implication whose all the premises are known
- for $L_{1,1} \wedge breeze \rightarrow B_{1,1}$ if we know $L_{1,1}$ and *breeze* then $B_{1,1}$ is added in knowledge base [4]

$$\begin{aligned} &P \Rightarrow Q \\ &L \wedge M \Rightarrow P \\ &B \wedge L \Rightarrow M \\ &A \wedge P \Rightarrow L \\ &A \wedge B \Rightarrow L \\ &A \\ &B \end{aligned}$$



- Applies Modus Ponens

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi}$$

- An and-or tree gets constructed

[4] $L_{1,1}$: location is [1,1]

Backward Chaining

- Works backward from query
- If query Q is known to be true, then no work is needed.
- Otherwise, find those implications whose conclusion is Q
- If all the premises of one of those implications can be proven true (by backward chaining) then Q is true

$$\begin{aligned} &P \Rightarrow Q \\ &L \wedge M \Rightarrow P \\ &B \wedge L \Rightarrow M \\ &\boxed{A \wedge P \Rightarrow L} \\ &A \wedge B \Rightarrow L \\ &A \\ &B \end{aligned}$$

- test(Q) is it true ?
- test(P) is it true ?
- test($L \wedge M$) ?
- ((test($A \wedge B$) or test($A \wedge P$)) and test($B \wedge L$) ? we know A and B so we have L this gives M
- Therefore P and hence Q

First Order Logic (Predicate Logic)

- We have **constants**, **variables**, **predicates** and **functions**
- Here $P(x)$ could means $\forall x$ we have $P(x)$ or $\exists x$ such that $P(x)$
- Variable x has a domain from where it gets values
- $\forall x, \exists y P(x, y)$ is not always same as $\exists y, \forall x P(x, y)$
- When we say $\exists$ a predicate then it is higher order logic

Examples

1. Not every customer have purchased milk and bread

$$\exists c \quad Cust(c) \wedge [\neg shop(milk, c) \vee \neg shop(bread, c)]$$

2. Only one customer have purchased guitar

$$\exists x \quad [Cust(x) \wedge shop(G, x) \wedge \forall y[\neg(x = y) \wedge Cust(y) \Rightarrow \neg shop(G, y)]]$$

3. Only one customer have purchased guitar and pen
4. Highest purchase in forenoon is more than afternoon.

Inference in First Order Logic

- **Universal Elimination** $\forall x \quad Feels(x, king)$ could be $Feels(Raju, king)$ substitution $\{x/\text{Raju}\}$ is done using some ground term.
- **Existential Elimination** $\exists x \quad Feels(x, king)$ could be $Feels(man, king)$ if *man* does not appear in knowledge base [5]
- **Existential Introduction** If $Feels(Raju, king)$ then we can say $\exists x \quad Feels(x, king)$

1. It is crime for Magadh to sell formula to a hostile country
2. Country Bhind, an enemy of Magadh have purchased some formula from Dara
3. Dara is from Magadh
4. Question: Is Dara a criminal?

[5] *man* is a name of person who feels like king

Prolog

- A logic programming language [6]
- Compile as ['a.pl'].
- If :- and , or ; not not
- write('hello'), nl

```
warm_blood(penguin).
warm_blood(human).
produce_milk(penguin).
produce_milk(human).
have_feather(penguin).
have_hair(human).
mammal(X) :-
warm_blood(X),
produce_milk(X),

have_hair(X).
```

```
?- mammal(penguin)
no
?- mammal(X).
X = human.
```

- ```
  is_even(X) :-
  Y is X//2, X =:= 2*Y.
  ```
- ```
  write('what is your name/ '), read(X), write('Hi '),write(X).
  ```

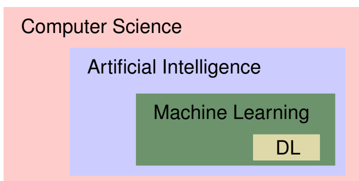
Many more things are possible

[6] http://www.swi-prolog.org/

Machine Learning

For some problems we don't precisely know either 1) how to solve, or 2) difficult to specify solution procedure

Then we go for **Machine Learning** (ML)



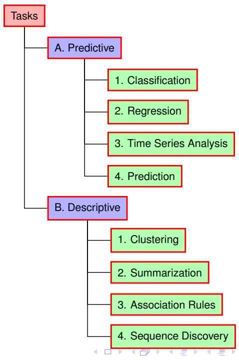
http://ktiwari.in/ml

Machine Learning: Tasks

Two broad categories of machine learning models are *Predictive* and *Descriptive*. Some of the related tasks are

Predictive
focuses towards new data item or expanding beyond known facts

Descriptive
focus to understand the available data



Types of Learning

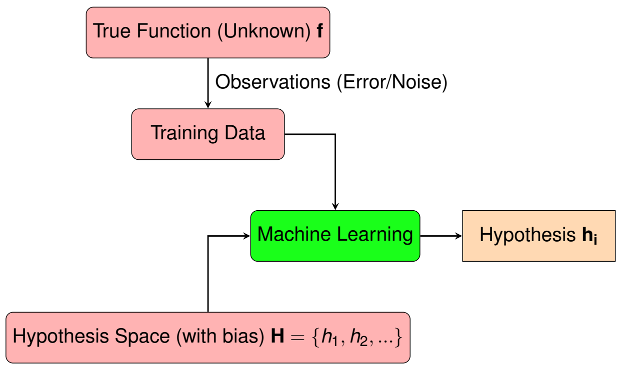
- **Supervised:** “right answers” are provided for sufficient training examples. Computer tells “right answers” for new input. Performance measure. (Classification and regression)
- **Unsupervised:** “right answers” are NOT provided and the computer tries to make sense of the data. How good the spread of items is. (clustering and association rule)
- **Semi-supervised:** “right answers” are provided for few training examples only
- **Active:** computer can ask questions. Needs less training. Opposite is passive learning
- **Lazy:** learner do not consolidate the findings.
- **Reinforced:** hit and trial method to minimize cost. (game playing)
- **Transfer:** Learning a task B to do A. (cycle riding for bike riding)
- **Deep:** processing like human brain

Applications of ML

In many domains including finance, robotics, bioinformatics, vision, natural language, *etc*.

- Spam filtering
- Speech/handwriting recognition
- Object detection/recognition
- Weather prediction
- Stock market analysis
- Search engines (e.g, Google)
- Ad placement on websites
- Adaptive website design
- Credit-card fraud detection
- Webpage clustering (e.g.,Google News)
- Machine Translation (e.g., Google Translate)
- Recommendation systems (e.g., Netflix, Amazon)
- Classifying DNA sequences
- Automatic vehicle navigation
- Performance tuning of computer systems
- Predicting good compilation flags for programs
- .. and many more

The Flow of ML



Probability of observing a dataset

Assume you are flipping a biased coin where $p(H) = 0.4$. What is the probability that you see this dataset $D =< H, H, T, T, H, H >$

- $p(H) = 0.4$
- $p(T) = 1 - p(H) = 1 - 0.4 = 0.6$
- If **all the trails are independent** then $p(D|\theta)$

$$= p(H) \times p(H) \times p(T) \times p(T) \times p(H) \times p(H)$$

$$= 0.4^4 \times 0.6^2 = 0.009216$$

Note: Order of elements in the data set do not matter in the trial. So $p(< H, H, H, H, T, T >)$ is same (in fact any other permutation)

What is θ
It is the parameter. For our case it represents $p(H) = 0.4$

Hypothesis

X	Y
10	0
11	0
12	0
13	1
14	0
15	1
16	0
17	1
18	1

h_1	h_2	...
0	1	...
0	0	...
0	1	...
1	0	...
1	1	...
1	0	...
1	1	...
1	0	...
1	1	...

- In this example h_1, h_2, ... are hypothesis.
- **Hypothesis** is a function that aims to provide value of the Y
- Can you identify h_1 and h_2
- Represent H as candidate set of hypothesis, *i.e.*$h_i \in H$
- Size of H is at least 2^m

Bayesian Learning

It is based on assumption that quantities of interest are governed by probability distribution

- Notation
  - $P(h)$: initial probability that hypothesis h holds
  - $P(D)$: probability that data D will be observed
  - $P(D|h)$: probability of observing data D given some world in which hypothesis h holds
  - $P(h|D)$: probability of holding hypothesis h when data D is observed

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Maximum a posteriori (MAP)

- Choose a hypothesis that maximizes $P(h|D)$

$$\begin{aligned} h_{MAP} &= \operatorname*{argmax}_{h \in H} P(h|D) \\ &= \operatorname*{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \operatorname*{argmax}_{h \in H} P(D|h)P(h) \end{aligned} \quad (2)$$

- Because $P(D)$ is independent of h
- If all the hypothesis are equally probable, we may further simplify called *maximum likelihood (ML)*

$$h_{ML} = \operatorname*{argmax}_{h \in H} P(D|h) \quad (3)$$

For our current example

X	Y
10	0
11	0
12	0
13	1
14	0
15	1
16	0
17	1
18	1

h_1	h_2	...
0	1	...
0	0	...
0	1	...
1	1	...
1	1	...
1	0	...
1	1	...
1	0	...
1	1	...

- Let bias for h_1 and h_2 be 2/50 and 6/50
- Since h_1 and h_2 are correct with probability 7/9 and 3/9 respectively
- Posterior is (7/9)*(2/50) and (3/9)*(6/50)
- Normalized probabilities are 0.4375 and 0.5625 respectively
- So MAP hypothesis corresponds to? h_2
- Can ML hypothesis? it is h_1
- **Brute-force MAP learning algorithm:** Evaluates posterior probability for all and returns the one with maximum
- **Consistent Learner:** learning algorithm is consistent learner if it provides a hypothesis that commits zero error

Thank You!

Thank you very much for your attention!

Queries ?

(Reference[7])

[7] 1) Book - *AIMA*, ch-07/08, Russell and Norvig. 2) Book - *Logic in CS*, ch-01/02, Mitchel Huth and Mark Ryan.