Introduction

Intelligent entity:
- Thinking Humanly
- Thinking Rationally
- Acting Humanly
- Acting Rationally

Birth of AI 1956

Searching problem

Search problem formulation

1. A finite set : Q
2. A set of initial states: $S \subseteq Q$
3. Operator or successor function $P(x)$
4. Goal test
5. Path cost

Criteria for evaluating searchs algorithms

1. Computational complexity
2. Space complexity
3. Completeness
4. Optimality

Basic search algorithms

General search algorithm

General idea : consider states, using successor functions

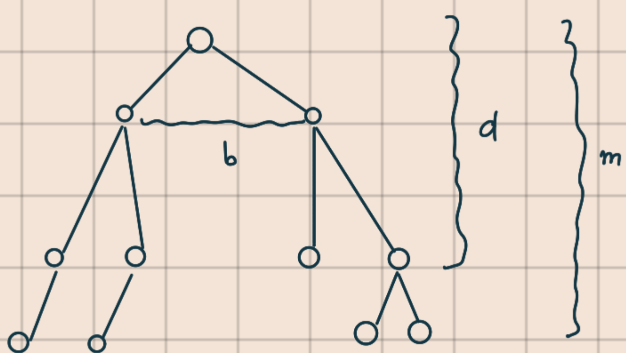
Expanding states creates a "search tree":
- Each state is a node
- Open node : node waiting for further expansion
- Expanded node : closed node.

General search algorithm

$Search(Q, S, G, P)$
(Q: state space, S: initial state, G: goals, P: successor function)
Input: search problem
Output: goal state (path to the goal state)
Initialize: $O \leftarrow S$ (O: the open node list)

while($O \neq \emptyset$) **do**
1. Select a node $n \in O$ and delete n from O
2. **if** $n \in G$, **return** (path to n)
3. Add $P(n)$ to O

return: no solution



Blind search

Breath-first search (BFS)

▸ **Principle**: among open nodes, choose the shallowest node (closest to the root) to expand

BFS algorithm :

$Search(Q, S, G, P)$
(Q: state space, S: initial state, G: goals, P: successor function)
Input: search problem
Output: goal state (path to the goal state)
Initialize: $O \leftarrow S$ (O: the open node list)

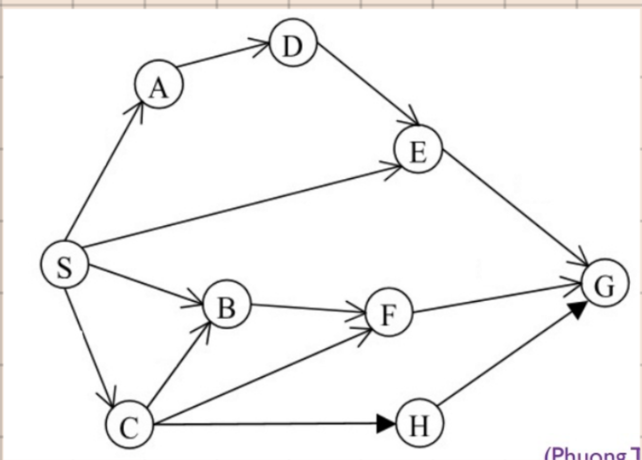
while($O \neq \emptyset$) **do**
1. take the first node n from O
2. **if** $n \in G$, **return** (path to n)
3. add $P(n)$ to the end of O

return no solution

Avoiding repeated nodes.

There can be multiple paths reaching to a node
- Can expand a node many times
- Can lead to an infinite loop

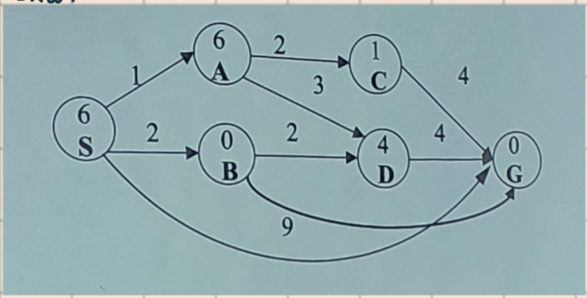
Ex1 :



No	Expanded node	Open node list
0		S
1	S	A_S, B_S, C_S, E_S
2	A_S	B_S, C_S, E_S, D_A
3	B_S	C_S, E_S, D_A, F_B
4	C_S	E_S, D_A, F_B, H_C
5	E_S	D_A, F_B, H_C, G_E
6	D_A	F_B, H_C, G_E
7	F_B	H_C, G_E
8	H_C	G_E
9	G_E	Goal

Path : $S \rightarrow E \rightarrow G$

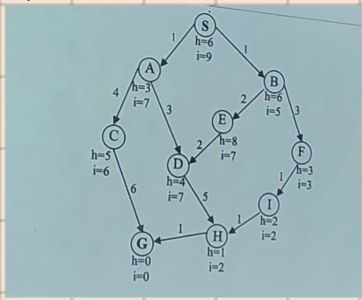
Ex2:



No	Expanded Node	Open node list
0		S
1	S	A_S, B_S, G_S
2	A_S	B_S, G_S, C_A, D_A
3	B_S	G_S, C_A, D_A
4	G_S	Goal

Path : $S \to G$

Ex3



No	Expanded Node	Open node list
0		S
1	S	A_S, B_S
2	A_S	B_S, C_A, D_A
3	B_S	C_A, D_A, E_B, F_B
4	C_A	D_A, E_B, F_B, G_C
5	D_A	E_B, F_B, G_C, H_D
6	E_B	F_B, G_C, H_D
7	F_B	G_C, H_D, I_F
8	G_C	Goal

Path : $S \to A \to C \to G$

Property of BFS

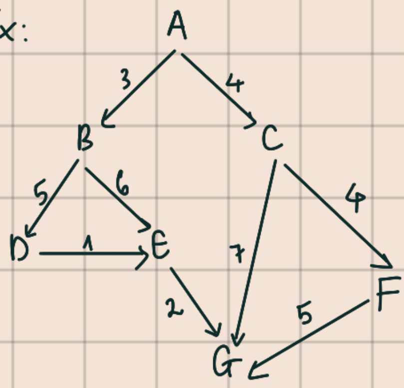
- Completeness : Yes
- Time : $1 + b + b^2 + \dots + b^d = O(b^d)$
- Space : $O(b^d)$
- Optimality :
  - Yes
  - Alway search all the nodes at the higher level before searching nodes at the lower level.

Nếu nút đã duyệt mà có chi phí tốt hơn thì lấy.

Uniform-cost search (UCF)

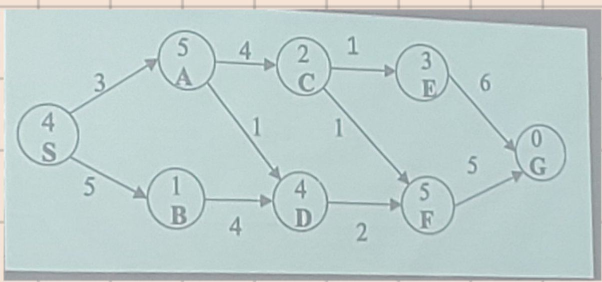
Principle: Choose the node with the smallest

Ex:



No	Expanded node	Set O
		A
	A	$B_A(3)$, $C_A(4)$
	B_A	$C_A(4)$, $D_B(8)$, $E_B(9)$
	C_A	$D_B(8)$, $E_B(9)$, $G_C(11)$, $F_C(8)$
	D_B	$E_B(9)$, $G_C(11)$, $F_C(8)$
	F_C	$E_B(9)$, $G_C(11)$
	E_B	$G_C(11)$
	G_C	Goal

Ex 2



No	Exp. node	Set O
		S
	S	$A_S(3)$, $B_S(5)$
	A_S	$B_S(5)$, $C_A(7)$, $D_A(4)$
	D_A	$B_S(5)$, $C_A(7)$, $F_D(6)$
	B_S	$C_A(7)$, $F_D(6)$
	F_D	$C_A(7)$, $G_F(11)$
	C_A	$G_F(11)$, $E_C(8)$
	E_C	$G_F(11)$
	G_F	Goal

Path: $G \leftarrow F \leftarrow D \leftarrow A \leftarrow S$

Cost = 11

No	Exp. Node	Set O
1		S
	S	A_S, B_S
	A_S	B_S, C_A, D_A
	B_S	C_A, D_A
	C_A	D_A, E_C, F_C
	D_A	E_C, F_C
	E_C	F_C, G_E
	F_C	G_E
	G_E	Goal

Path: $G \leftarrow E \leftarrow C \leftarrow A \leftarrow S$

Depth First Search (DFS)

Principle: among open nodes, choose the deepest node to expand.

DFS Algorithm

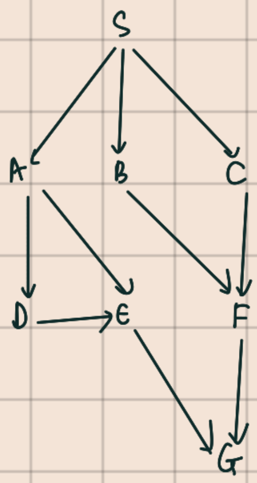
$Search(Q, S, G, P)$
(Q: state space, S: initial state, G: goals, P: successor function)

Input: search problem
Output: goal state (path to the goal state)
Initialize: $O \leftarrow S$ (O: the open node list)

while($O \neq \emptyset$) **do**
1. take the first node n from O
2. **if** $n \in G$, **return** (path to n)
3. add $P(n)$ to the head of O

return no solution

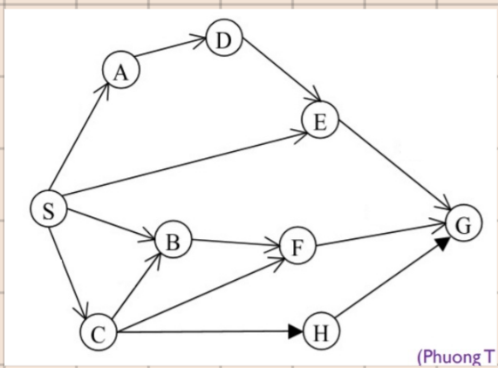
$\boxed{\text{Ex1}}$:



No	Exp node	Set O
		S
1	S	A_S, B_S, C_S
2	A_S	D_A, E_A, B_S, C_S
3	D_A	E_D, E_A, B_S, C_S.
4	E_D	G_E, E_A, B_S, C_S.
5	G_E	Goal.

Path: $G \leftarrow E \leftarrow D \leftarrow A \leftarrow S$.

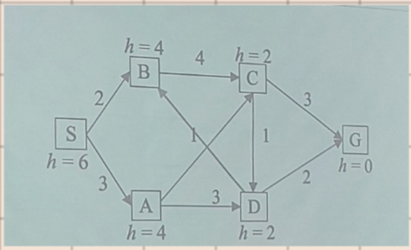
Ex2



No	Exp. node	Set O
		S
	S	A_S, B_S, C_S, E_S
	A_S	D_A, B_S, C_S, E_S
	D_A	E_D, B_S, C_S, E_S
	E_D	G_E, B_S, C_S, E_S
	G_E	Goal

Path: $S \rightarrow A \rightarrow D \rightarrow E \rightarrow G$

Ex2



No	Exp. Node	Set O
		S
	S	A_S, B_S
	A_S	C_A, D_A, B_S
	C_A	D_C, G_C, B_S
	D_C	G_D, G_C, B_S
	G_D	Goal

Path: $S \to A \to C \to D \to G$

Property:

- **Completeness?**
  - No: when the depth of state space is infinite
- **Optimality?**
  - No
- **Time?**
  - $O(b^m)$: very large if m is greater than d
  - If there are many solutions, DFS can be much faster than BFS
- **Space?**
  - $O(bm)$: much better than BFS

Iterative deepening search (IDS)

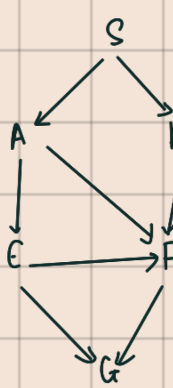
Principle: use DFS but never extend nodes with depth beyond a certain limit. The depth limit will be gradually increased until a solution is found.

IDS Algorithm

Search(Q, S, G, P)
(Q: state space, S: initial state, G: goals, P: successor function)
Input: search problem
Output: goal state (path to the goal state)
Initialize: $O \leftarrow S$ (O: the open node list)
$c = 0$ (current depth)

while (1) **do**
1. **while** ($O \neq \emptyset$) **do**
   a. take the first node n from O
   b. **if** $n \in G$, **return** (path to n)
   c. **if** $depth(n) < c$ **then**
      add $P(n)$ to the head of O
2. $c++$; $O = S$

Ex1:



C = 0

0		S
1	S	$\emptyset$

C = 1

0		S
1	S	A_S, B_S
2	A_S	B_S
3	B_S	$\emptyset$

$C=2$

0		S
1	S	A_S, B_S
3	A_S	E_A, F_A, B_S
4	E_A	F_A, B_S
5	F_A	B_S
6	B_S	$\emptyset$

$C=3$

0		S
1	S	A_S, B_S
2	A_S	E_A, F_A, B_S
3	E_A	F_E, G_E, F_A, B_S
4	F_E	G_E, F_A, B_S
5	G_E	Goal

Path: $S \to A \to E \to G$

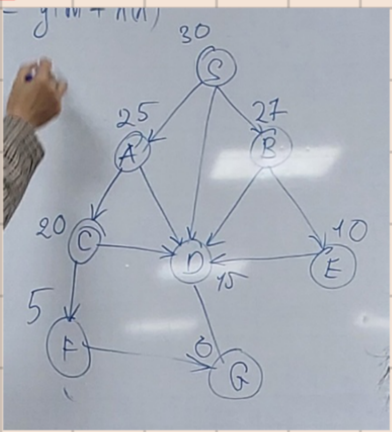
Informed search: $f(n) = \underbrace{g(n)}_{S \to n} + \underbrace{h(n)}_{\text{heuristics}}$

Greedy Search

Principle: expand the node with the cheapest path to a goal first

$f(n) = h(n)$: heuristic function, estimate cost of the path from n to a goal node.

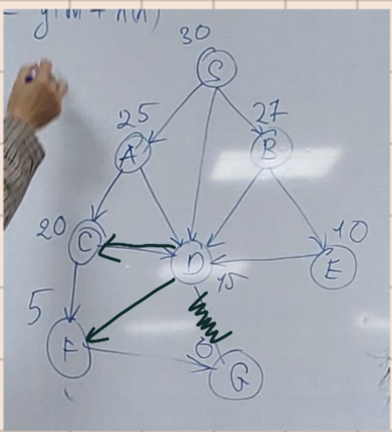
Ex1



	Exp node	Set O
0	$\emptyset$	S
1	S	$A_S(25)$, $B_S(27)$, $D_S(15)$
2	D_S	$A_S(25)$, $B_S(27)$, $G_D(0)$
3	G_D	Goal

Path $S \to D \to G$

Ex2:



0	$\emptyset$	S
1	S	$A_S(25)$, $B_S(27)$, $D_S(15)$
2	D_S	$A_S(25)$, $B_S(27)$, $F_D(5)$, $C_D(20)$
3	F_D	$A_S(25)$, $B_S(27)$, $C_D(20)$, $G_F(0)$
4	G_F	Goal

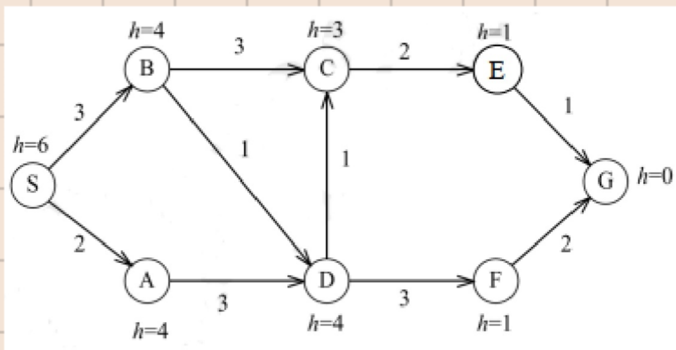
Path: $G \leftarrow F \leftarrow D \leftarrow S$

Properties:

- Completeness?
  - No (can have a loop, or have a branch of infinite nodes with small value of function h but do not lead to a goal)
- Optimality?
  - No
- Time?
  - $O(b^m)$
  - If the *heuristic* function is good, the algorithm can be much faster
- Space?
  - $O(b^m)$: store all nodes in the memory
  - If the *heuristic* function is good, the number of nodes to store can be reduced significantly

Ex3

	Exp. node	Set O
0	∅	S
1	S	$A_S(4)$, $B_S(4)$
2	A_S	$B_S(4)$, $D_A(4)$
3	B_S	$D_A(4)$, $C_B(3)$
4	C_B	$D_A(4)$, $E_C(1)$
5	E_C	$D_A(4)$, $G_E(0)$
6	G_E	Goal



Path: $G \Leftarrow E \Leftarrow C \Leftarrow B \Leftarrow S$

A^* search

Method: $f(n) = g(n) + h(n)$

↳ $h(n)$: heuristic function, cost from n to a goal node

↳ $g(n)$: cost so far to reach n

↳ $f(n)$: total cost

Algorithm

$A*(Q, S, G, P, c, h)$

(Q: state space, S: initial state, G: goals, P: successor function, c: cost, h: heuristic)

Input: search problem, heuristic function h

Output: goal state (path to the goal state)

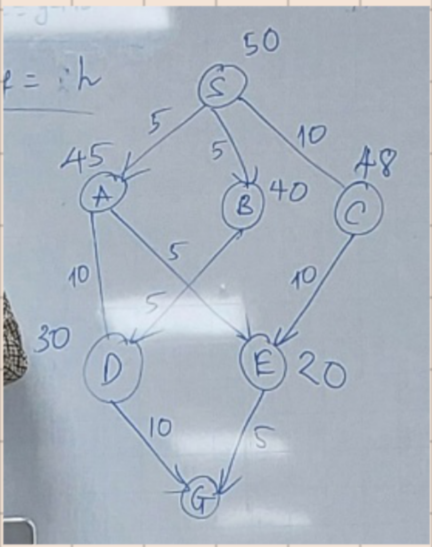
Initialize: $O \leftarrow S$ (O: the open node list)

while ($O \neq \emptyset$) **do**

1. take node n whose $f(n)$ is the smallest from O
2. **if** $n \in G$, **return** (path to n)
3. for each $m \in P(n)$
   a) $g(m) = g(n) + c(n, m)$
   b) $f(m) = g(m) + h(m)$
   c) add m along with $f(m)$ to O

return no solution

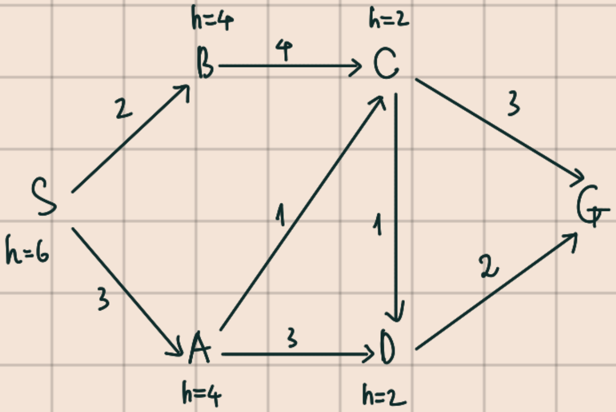
Ex1



	Exp. node	Set O
0	∅	S
1	S	$A_S(50)$, $B_S(45)$, $C_S(58)$
2	B_S	$A_S(50)$, $C_S(58)$, $D_B(40)$
3	D_B	$A_S(50)$, $C_S(58)$, $G_D(20)$
4	G_D	Goal

Path: $S \to B \to D \to G$

Ex2



	Exp. node	Set O
0	∅	S
1	S	$A_S(7)$, $B_S(6)$
2	B_S	$A_S(7)$, $C_B(8)$
3	A_S	$C_A(6)$, $D_A(8)$
4	C_A	$D_C(7)$, $G_C(7)$
5	D_C	$G_C(7)$
6		
7	G_C	Goal

Path: $S \to A \to C \to G$

Property

- Complete?
  - Yes (unless there are infinite nodes n with $f(n) \le f(G)$)
- Optimality?
  - Yes (if heuristic function h is admissible)
- Time?
  - $O(b^m)$
  - If the *heuristic* function is good, the algorithm can be much faster
- Space?
  - $O(b^m)$: store all nodes in the memory
  - If the *heuristic* function is good, the number of nodes to store can be reduced significantly

- Admissible heuristic function
  - For every node n, $h(n) \le h^*(n)$, where $h^*(n)$ is the true cheapest cost from n to a goal node
  - Example: straight-line distance is an admissible heuristic function

- Heuristic functions are constructed depending on each specific problem
  - A problem may have some heuristic functions
  - The quality of the heuristic function greatly affects the search process

Iterative deepening A* – IDA*

Điểm giống duy nhất với A*: $f = g + h$

IDA*: là DFS bị giới hạn bởi f.

Algorithm

$IDA^*(Q, S, G, P, c, h)$

Input: search problem, heuristic function h

Output: goal state (path to the goal state)

Initialize: $O \leftarrow S$ (O: the open node list)

Threshold $i \leftarrow 0$

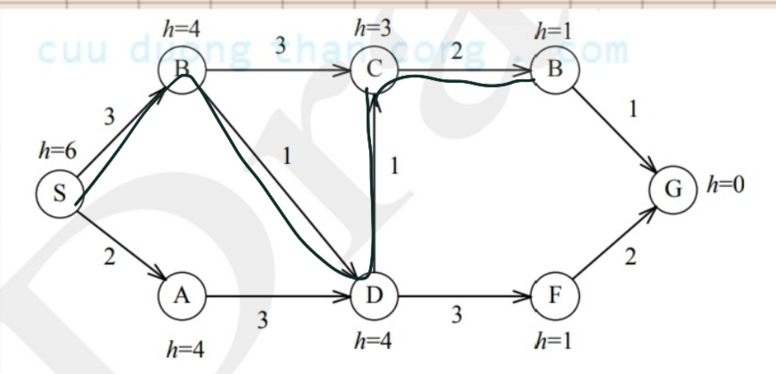
while (1) **do**

1. **while** ($O \neq \emptyset$) **do**
   a) Take the first node n from O
   b) **if** $n \in G$, **return** (path to n)
   c) For each $m \in P(n)$
      i) $g(m) = g(n) + c(m, n)$
      ii) $f(m) = g(m) + h(m)$
      iii) **if** $f(m) \leq i$ **then** add m to the head of O
2. $i \leftarrow i + \beta, O \leftarrow S$

Property

- Completeness?
  - Yes
- Optimality?
  - β-optimal (cost of the found solution does not exceed β compared to cost of the optimal solution)
- Time?
  - Computational complexity is greater than that of A* search
- Space?
  - Requires linear memory

Ex1



$i = 0 \rightarrow \emptyset$

$i = 2 \rightarrow \emptyset$

$i = 4 \rightarrow \emptyset$

i = 6:

0	$\cancel{\emptyset}$	S
1	S	A_S (6)
2	A_S	$\cancel{\emptyset}$

i=8

0	$\cancel{\emptyset}$	S
1	S	$A_S(6)$, $B_S(7)$
2	A_S	$B_S(7)$
3	B_S	$D_B(8)$
4	D_B	$C_D(8)$, $F_D(8)$
5	C_D	$E_C(8)$, $F_D(8)$
6	E_C	$G_E(8)$, $F_D(8)$
7		
8	G_E	Goal

Path: $G \leftarrow E \leftarrow C \leftarrow D \leftarrow B \leftarrow S$

When to add repeated nodes to the open node list?

- Greedy
  - **No**: adding repeated nodes does not change the algorithm (may lead to loops)
- A*
  - In cases the repeated node has better cost, it will be added to the list (if it is already expanded) or updated to replace the old node (if it is on the list)
- IDA*:
  - **Yes**

Test:

(8)

0	$\emptyset$	S
1	s	d_s, e_s, p_s
2	d_s	e_s, p_s, b_d, c_d
3	e_s	p_s, b_d, c_d, h_e, r_e
4	p_s	b_d, c_d, h_e, r_e, q_p
5	b_d	c_d, h_e, r_e, q_p, a_b
6	c_d	h_e, r_e, q_p, a_b
7	h_e	r_e, q_p, a_b
8	r_e	q_p, a_b, f_r
9	q_p	a_b, f_r
10	a_b	f_r
11	f_r	G_f
12	G_f	Goal

Path: Goal $\leftarrow f \leftarrow r \leftarrow e \leftarrow s$

(9)

0	$\emptyset$	S
	s	$d_s(3)$, $e_s(9)$, $p_s(1)$
	p_s	$d_s(3)$, $e_s(9)$, $q_p(16)$
	d_s	$q_p(16)$, $b_d(4)$, $c_d(11)$, $e_d(5)$
	b_d	$q_p(16)$, $c_d(11)$, $e_d(5)$, $a_b(6)$
	e_d	$q_p(16)$, $c_d(11)$, $a_b(6)$, $h_e(6)$, $r_e(14)$
	a_b	$q_p(16)$, $c_d(11)$, $h_e(6)$, $r_e(14)$
	h_e	$q_h(10)$, $c_d(11)$, $r_e(14)$
	q_h	$c_d(11)$, $r_q(13)$
	c_d	$r_q(13)$
	r_q	$f_r(18)$
	f_r	Goal (23)

Path: $G \leftarrow f \leftarrow r \leftarrow q \leftarrow h \leftarrow e \leftarrow d \leftarrow s$

Cost: 23

(10)

0	$\emptyset$	S
1	S	$A_S(6)$, $B_S(7)$
2	A_S	$B_S(7)$, $C_A(9)$
3	B_S	$C_B(8)$, $D_B(11)$
4	C_B	$D_B(10)$, $E_C(8)$
5	E_C	$D_B(10)$, $G_E(9)$
6	G_E	Goal
7		

Path: $G \leftarrow E \leftarrow C \leftarrow B \leftarrow S$

Local search

local search problem:

- State space X
- Objective function $Obj: X \to R$
- Set of actions (to generate neighbors)
  - $N(x)$ set of neighbors of x
- Task: Find state x^* such that $Obj(x^*)$ is maximum (or minimum)

Hill-climbing search

Principle: from the current state, consider the set of neighbors, move to a better state.

Goal state: stop when there is no better neighbor. → can find global max / local max.

Algorithm:

Input: combinatorial optimization problem

Output: state with the maximum value of the objective function (local max)

1, Select a random state x

2, let Y denotes the set of neighbors of x

3, if $\forall y_i \in Y$: $Obj(y_i) < Obj(x)$

return x

4, $x \leftarrow y_i$ where $i = \arg\max_i (Obj(y_i))$

5, Go to 2.

Random hill-climbing

Algorithm:

1, Select a random state x

2, let Y denote the set of neighbors of x

3, Select a random state $y_i \in Y$

4, if $Obj(y_i) > Obj(x)$

$x \leftarrow y_i$

5, Go to 2 if still have patience

Propositional logic

knowledge: in4 đc tổng hợp, khái quát và có tính đúng đắn

knowledge representation language = syntax + semantics + reasoning problem

→ có 1 cách hiểu duy nhất

Propositional logic

Syntax:

Symbol:
- Truth: True / False
- Propositional (variable): P, Q, ...
- logic connective: $\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$
- Brackets (and)

Syntactic Rule:
- Conjunction $\wedge$
- Disjunction $\vee$
- Negation $\neg$
- Implication $\Rightarrow$
- Equivalence $\Leftrightarrow$

$A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n$: clausal sentences.

Semantics:

Interpretation: a way of assigning each propositional variable a truth value

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	True	True	False	True	True	False
False	False	True	False	False	True	True

Satisfiable: $\exists$ an interpretation that make the formula true.

ex: $(P \wedge Q) \vee \neg R$

Unsatisfiable: A formula whose truth table contains only false

ex: $P \wedge \neg P$

Valid: true in every interpretation

ex: $P \vee \neg P$

Model: an interpretation that makes the formula true

ex: $P \leftarrow$ true, $Q \leftarrow$ false, $R \leftarrow$ false

Logical equivalences:

Basic LE:
- Implication: $A \Rightarrow B \equiv \neg A \vee B$
- Equivalent: $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
- Double negation: $\neg(\neg A) \equiv A$

De Morgan's
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$

Commutative
- $A \vee B \equiv B \vee A$
- $A \wedge B \equiv B \wedge A$

Associative

$(A \vee B) \vee C \equiv A \vee (B \vee C)$

$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

Distributive

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Conjuntive normal form (CNF) : chuẩn tắc hội

$(A \vee E \vee F \vee G) \wedge (B \vee C \vee D) \rightarrow$ tách câu

Convert: $\Leftrightarrow$, $\Rightarrow$, de Morgan, phân phối.

1) $(P \wedge Q) \Rightarrow P \equiv \neg(P \wedge Q) \vee P \equiv \neg P \vee \neg Q \vee P = 1 \vee \neg Q = T$

3) $\neg P \Rightarrow (P \Rightarrow Q) \equiv P \vee (\neg P \vee Q) = T \vee Q = T$

4) $(P \wedge Q) \Rightarrow (P \Rightarrow Q) \equiv \neg(P \wedge Q) \vee (\neg P \vee Q) \equiv \neg P \vee \neg Q \vee \neg P \vee Q = T$

5) $(P \Leftrightarrow Q) \equiv (P \land Q) \lor (\neg P \land \neg Q)$

$$VT \equiv (P \Rightarrow Q) \land (Q \Rightarrow P) \equiv (\neg P \lor Q) \land (\neg Q \lor P)$$

$$\equiv [(\neg P \lor Q) \land \neg Q] \lor [(\neg P \lor Q) \land P]$$

$$\equiv (\neg P \land \neg Q) \lor \cancel{(Q \land \neg Q)} \lor \cancel{(\neg P \land P)} \lor (Q \land P) = VP$$

6) $(P \Rightarrow Q) \lor \neg(R \lor \neg S)$

$$\equiv (\neg P \lor Q) \lor (\neg R \land S) \equiv (\neg P \lor Q \lor \neg R) \land (\neg P \lor Q \lor S)$$

Logical inference:

- A formula H is said to be a logical consequence of a set of formulas $G = \{G_1, \ldots, G_m\}$ if in any interpretation that G is true then H is also true
- An inference procedure consists of a set of premises and a conclusion

$$\frac{\textit{set of premises}}{\textit{conclusion}} \quad \begin{array}{l} \rightarrow ĐK \\ \rightarrow HQ \end{array}$$

  - Soundness: if conclusion is a logical consequence of the set of premises
  - Completeness: if can find every logical consequence of the set of premises
- Notations
  - KB : Knowledge Base, set of known formulas
  - KB $\vdash \alpha$: α is a logical consequence of KB

Inference using truth table: ex: $KB = \{A \lor C,\ B \lor \neg C\}$

$$\alpha = A \lor B$$

A	C	B	¬C	A ∨ C	B ∨ ¬C	(A∨C) ∧ (B∨¬C)	A ∨ B
T	T	T	F	✓ T	✓ T	✓ T	T
T	F	T	T	✓ T	✓ T	✓ T	T
F	T	T	F	✓ T	✓ T	✓ T	T
F	F	T	T	F	T	F	T
T	T	F	F	T	F	F	T
T	F	F	T	✓ T	✓ T	✓ T	T
F	T	F	F	T	F	F	F
F	F	F	T	F	T	F	F

- Modus Ponens $\dfrac{\alpha \Rightarrow \beta,\ \alpha}{\beta}$
- Modus Tollens $\dfrac{\alpha \Rightarrow \beta,\ \neg\beta}{\neg\alpha}$
- And-Elimination $\dfrac{\alpha_1 \wedge \ldots \wedge \alpha_i \wedge \ldots \wedge \alpha_m}{\alpha_i}$
- And-Introduction $\dfrac{\alpha_1, \ldots, \alpha_i, \ldots, \alpha_m}{\alpha_1 \wedge \ldots \wedge \alpha_i \wedge \ldots \wedge \alpha_m}$

- Or-Introduction $\dfrac{\alpha_i}{\alpha_1 \vee \ldots \vee \alpha_i \vee \ldots \vee \alpha_m}$
- Double-Negation Elimination $\dfrac{\neg(\neg\alpha)}{\alpha}$
- Transitivity $\dfrac{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma}{\alpha \Rightarrow \gamma}$
- Unit Resolution $\dfrac{\alpha \vee \beta, \neg\beta}{\alpha}$
- Resolution $\dfrac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$

ex: $\overset{(1)}{(Q \wedge S \Rightarrow G \wedge H)}$ $\overset{(2)}{(P \Rightarrow Q)}$ $\overset{(3)}{(R \Rightarrow S)}$ $\overset{(4)}{P}$ $\overset{(5)}{R}$

Use the Modus Ponens for (2) & (4) : Q (6)
Use the Modus Ponens for (3) & (5) : S (7)
Use the And Introduction for (6) & (7) : $\dfrac{Q \wedge S}{\alpha}$ (8)
Use the Modus Ponens for (8) & (1) :
$G \wedge H$ (9)
Use the And Elimination for (9) : G (10).

(10) is logical consequense of KB.

Predicate logic

Characteristics of predicate logic:
- propositional logic
- predicate logic : objects, properties, relationships, function

Syntax
- function: f, g, sin, cos, mother, ...
- constant: An, a, b, ...
- variable : $x, y, z, \ldots$
- predicate : P, Q, like, Friend, ...
- logical connective: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$, $\underbrace{\forall, \exists}_{\text{quantifier symbols}}$

logical equivalences

1. $\forall x G(x) \equiv \forall y G(y)$
2. $\exists x G(x) \equiv \exists y G(y)$
3. $\neg(\forall x G(x)) \equiv \exists x(\neg G(x))$
4. $\neg(\exists x G(x)) \equiv \forall x(\neg G(x))$
5. $\forall x(G(x) \land H(x)) \equiv \forall x G(x) \land \forall x H(x)$
6. $\exists x(G(x) \lor H(x)) \equiv \exists x G(x) \lor \exists x H(x)$

ex : 4, $\forall x$ (Nấm(x) $\land$ Đỏ(x) $\Rightarrow$ Độc(x))

5, $\forall x$ (Nấm(x) $\land$ Đỏ(x) $\Rightarrow \neg$ Độc(x))

Inference rule

Substitution

(thế)

- **Notation**: $SUBST(\theta, \alpha)$
- **Meaning**: substitute value θ into formula α
- Example
  - $SUBST(\{x/Nam, y/An\}, Like(x,y)) = Like(Nam, An)$

Universal elimination

$$\frac{\forall x\ \alpha}{SUBST(\{x/g\}, \alpha)}$$

Example:

$\forall x\ Like(x, IceCream) \xrightarrow{\{x/Nam\}} Like(Nam, IceCream)$

Existential elimination

$$\frac{\exists x\ \alpha}{SUBST(\{x/k\}, \alpha)}$$ k has not appeared in KB

k: Skolem constant.

Example:

$\exists x\ GoodAtMath(x) \xrightarrow{\{x/C\}} GoodAtMath(C)$

Existential introduction

$$\frac{\alpha}{\exists x\ SUBST(\{g/x\}, \alpha)}$$

Example:

$Like(Nam, IceCream) \xrightarrow{\{Nam/x\}} \exists x\ Like(x, IceCream)$

ex:

Bob là trâu		
Pat là lợn		
Trâu to hơn lợn		
Bob to hơn Pat?		

(1) Buffalo (Bob)

(2) Pig (Pat)

(3) $\forall x, y$ Buffalo(x) $\land$ Pig(y) $\Rightarrow$ Bigger(x, y)

And-Introduction (1)(2)	$Buffalo(Bob) \land Pig(Pat)$	(4)
Universal elimination (3)	$Buffalo(Bob) \land Pig(Pat) \Rightarrow Bigger(Bob, Pat)$	(5)
Modus Ponens, (4)(5)	$Bigger(Bob, Pat)$	

Unification

- **Notation**: $UNIFY(p, q) = (\theta)$
  $SUBST(\theta, p) = SUBST(\theta, q)$
  θ is called unifier

General Modus Ponens (GMP)

- Suppose that we have atoms p_i, p_i', q, and substitution θ such that $UNIFY(p_i, p_i') = \theta$, for all i
- Then:

$$\frac{p_1', p_2', \ldots, p_n', (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q)}{SUBST(\theta, q)}$$

ex: $Buff(Bob)$ $\quad p_1'$ $\quad (1)$

$Pig(Pat)$ $\quad p_2'$ $\quad (2)$

$\forall x, y \quad \underbrace{Buff(x)}_{p_1} \land \underbrace{Pig(y)}_{p_2} \Rightarrow Bigger(x, y) \quad (3)$

GMP (1), (2), (3) $\quad \theta = \{x/Bob, y/Pat\}$

$Subst(\theta, q) = Bigger(Bob, Pat)$

Forward chaining

- When new formula p is added to KB:
  - For each rule q that p can be unified with a left-hand side part:
    - If the remaining parts of the left hand side already exist, add the right-hand side to KB and continue

ex:

$$KB \begin{cases} \forall x \quad Cat(x) \Rightarrow Like(x, Fish) & (1) \\ \forall x, y \; Cat(x) \land Like(x, y) \Rightarrow Eat(x, y) & (2) \\ Cat(Tom) & (3) \end{cases}$$

GMP: (1), (3) $\xrightarrow{x/Tom}$ $Like(Tom, Fish)$ (4)

GMP: (3), (4), 2 $\xrightarrow{x/Tom,\; y/Fish}$ $Eat(Tom, Fish)$ (5)

ex

Cho các mệnh đề sau dưới dạng ngôn ngữ tự nhiên
- Tất cả những người đi học là người có văn hoá.
- Trộm không có văn hoá.
- Một số tên trộm thông minh.

a) Viết các câu trên dưới dạng logic vị từ
b) Viết câu truy vấn sau "Có một số người thông minh không được đi học" dưới dạng lôgic vị từ sử dụng các vị từ đã cho ở trên và chứng minh câu truy vấn là đúng sử dụng suy diễn tiến.

a) (1) $\forall x \quad Person(x) \land Gotoschool(x) \Rightarrow Educational(x)$

(2) $\forall x \quad Thief(x) \Rightarrow \neg Educational(x)$

(3) $\exists x \quad Thief(x) \land Intelligent(x)$ $\qquad \exists x \; Intelligent(x) \land \neg Gotoschool(x)$

Backward chaining

- For question q, if exists q' that can be unified with q, return unifier
- For each rule with the right-hand side q' that can be unified with q, try to prove the left-hand side parts using backward chaining

Reasoning using resolutions

- Resolution in predicate logic
  - Given two formulas, where P_i and Q_i are literals
    - $P_1 \vee P_2 \vee \ldots \vee P_n$
    - $Q_1 \vee Q_2 \vee \ldots \vee Q_m$
  - If P_j and $\neg Q_k$ can be unified by θ, we have the following resolution

$$\frac{P_1 \vee P_2 \vee \ldots \vee P_n, Q_1 \vee Q_2 \vee \ldots \vee Q_m}{SUBST(\theta, P_1 \vee \ldots \vee P_{j-1} \vee P_{j+1} \vee \ldots \vee P_n \vee Q_1 \vee \ldots \vee Q_{k-1} \vee Q_{k+1} \vee \ldots \vee Q_m)}$$

Example:

$$\frac{\forall x(Rich(x) \vee Good(x)), \neg Good(Nam) \vee Handsome(Nam)}{Rich(Nam) \vee Handsome(Nam)}$$

Resolution and Reductio ad absurdum

- Need to prove $KB \vdash Q$?
- Method:
  - Add $\neg Q$ to KB, prove that there exists a subset of KB that has False value
  - $(KB \vdash Q) \Leftrightarrow (KB \wedge \neg Q \vdash False)$

Algorithm

- $KB = UNION(KB, \neg Q)$
- **while** (KB does not contain False) **do**
  - 1. Select two formulas S_1, S_2 from KB so that we can apply resolution
    - Add the result of the resolution to KB
  - 2. If does not exist two such formulas
    - **return** False
- **end while**
- **return** Success

ex:

KB:
$\neg A \vee \neg B \vee P$ (1)
$\neg C \vee \neg D \vee P$ (2)
$\neg E \vee C$ (3)
A (4)
E (5)
D (6)
Prove that: $KB \vdash P$

Add (7) $\neg P$ to KB.

(2) & (7) $\Rightarrow$ $\neg C \vee \neg D$ (8)

(8) & (6) $\Rightarrow$ $\neg C$ (9)

(9) & (3) $\Rightarrow$ $\neg E$ (False)

$\Rightarrow$ $KB \vdash P$

Conjunctive Normal Form (CNF) and Clause form

- Step 1: Eliminate equivalences
  - Replace $P \Leftrightarrow Q$ by $(P \Rightarrow Q) \land (Q \Rightarrow P)$
- Step 2: Eliminate implications
  - Replace $P \Rightarrow Q$ by $\neg P \lor Q$
- Step 3: Move negations close to predicates
  - Apply De Morgan's laws and replace $\neg(\neg A)$ by A:

    $\neg(\neg P) \equiv P$

    $\neg(P \land Q) \equiv \neg P \lor \neg Q$

    $\neg(P \lor Q) \equiv \neg P \land \neg Q$

    $\neg(\forall x Q) \equiv \exists x(\neg Q)$

    $\neg(\exists x Q) \equiv \forall x(\neg Q)$
- Step 4: Standardize variable names so that each quantifier has its own variable
  - Example

    $\boxed{\begin{array}{l} \forall x \neg P(x) \lor Q(x) \\ \forall x \neg R(x) \lor Q(x) \end{array}} \Longrightarrow \boxed{\begin{array}{l} \forall x \neg P(x) \lor Q(x) \\ \forall \mathbf{y} \neg \mathbf{R}(\mathbf{y}) \lor \mathbf{Q}(\mathbf{y}) \end{array}}$
- Step 5: Eliminate existential quantifiers by using Skolem constants and Skolem functions
  - Replace $\exists x\, P(x)$ by $P(C)$, where C is a new constant (Skolem)
  - If $\exists$ is inside $\forall$, replace by a new function whose variable is a variable of $\forall$ (Skolem function)
  - Example:

  Replace $\forall x\, \exists y\, P(x,y)$ by $\forall x\, P(x, f(x))$, $f(x)$ is a Skolem function
- Step 6: Eliminate universal quantifiers ($\forall$)
  - Move universal quantifiers to the left-hand side and remove them
  - Example: transform $\forall x\, (P(x,y) \lor Q(x))$ into $(x,y) \lor Q(x)$
- Step 7: Apply distributive laws
  - $(P \land Q) \lor R \equiv (P \lor R) \land (Q \lor R)$
  - $(P \lor Q) \lor R \equiv (P \lor Q \lor R)$
- Step 8: Eliminate conjunctions
  - Eliminate conjunctions to from clauses
  - Example: transform $(P \lor R \lor S) \land (Q \lor \rceil R)$ into two formulas: 1) $P \lor R \lor S$ 2) $Q \lor \rceil R$
- Step 9: Standardize variable names so that each formula has its own variables

ex: $\forall x (P(x) \Rightarrow \forall y (P(y) \Rightarrow P(f(x,y)) \land \neg \forall y(Q(x,y) \Rightarrow P(y)))$

$\forall x (\neg P(x) \lor \forall y (\neg P(y) \lor P(f(x,y)) \land \neg \forall y(\neg Q(x,y) \lor P(y)))$

$\forall x (\neg P(x) \lor \forall y (\neg P(y) \lor P(f(x,y)) \land \exists y (Q(x,y) \lor \neg P(y)))$

$\forall x (\neg P(x) \lor \forall y (\neg P(y) \lor P(f(x,y)) \land \exists z (Q(x,z) \land \neg P(z)))$

$\forall x (\neg P(x) \lor \forall y (\neg P(y) \lor P(f(x,y)) \land Q(x, g(x)) \land \neg P(g(x))$

$(\neg P(x) \lor (\neg P(y) \lor P(f(x,y)) \land Q(x, g(x)) \land \neg P(g(x))$

8. 1) $\neg P(x) \lor \neg P(y) \lor P(f(x,y))$ 9,

2, $\neg P(x) \lor Q(x, g(x))$

3, $\neg P(x) \lor \neg P(g(x))$

ex

- Cho các câu sau
1. Mọi bé trai đều thích chơi bóng đá
2. Ai thích chơi bóng đá đều có giày đá bóng
3. Nam là một bé trai

Câu hỏi
a) Biểu diễn các câu trên ở dạng logic vị từ
b) Chuyển các câu logic vị từ vừa viết về dạng chuẩn tắc hội
c) Viết câu truy vấn "Nam có giày đá bóng" dưới dạng logic vị từ và chứng minh sử dụng phép giải

1, $\forall x \quad Boy(x) \Rightarrow like(x, pfootball)$

2, $\forall x \quad like(x, pfootball) \Rightarrow Hasshose(x)$

3, $Boy(Nam)$.

CNF.

1, $\forall x \quad \neg Boy(x) \vee like(x, pfootball)$

$\neg Boy(x) \vee like(x, pfootball)$

2, $\neg like(x, pfootball) \vee hasshose(x)$

3: CNF

Haveshoes(Nam)?

KB (4) — (4),(3) — Like(Nam, pfootball) (6)

(5) — (5),(6) — Hasshose(Nam)

(6)

ex:

- Giả sử ta biết các thông tin sau
1. Ông Ba nuôi một con chó
2. Hoặc ông Ba hoặc ông Am đã giết con mèo Bibi
3. Mọi người nuôi chó đều yêu động vật
4. Ai yêu quý động vật cũng không giết động vật
5. Chó mèo đều là động vật

Hỏi ai đã giết con mèo Bibi?

1, $\exists x \quad Dog(x) \wedge Raise(Ba, x)$

2, $Cat(Bibi) \wedge (Kill(Ba, Bibi) \vee Kill(Am, Bibi))$

Probabilistic inference

Inference with uncertain evidence

Approaches:
- Multivalued logic
- Fuzzy logic
- Possibility theory
- Probabilistic inference

Principle of probabilistic inference

Instead of inferring about the "true" or "false" of a proposition (2 values), inferring about the "belief" that the proposition is true or false (infinite)

- Assign a belief value to each proposition
- Express the belief value as a probability value; using probability theory to work with this value
- For proposition A
  - Assign a probability $P(A)$: $0 \leq P(A) \leq 1$;
  - $P(A) = 1$ if A is true, $P(A) = 0$ if A is false
- Example:
  - $P(Cold = true) = 0.6$
  - the patient has a cold with a probability of 60%, "Cold" is a random variable that can receive True or False

Axioms of probability and some basic properties

Axioms of probability

1. $0 \leq P(A = a) \leq 1$ for all a in the value domain of A
2. $P(True) = 1$, $P(False) = 0$
3. $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

Basic properties

1. $P(\neg A) = 1 - P(A)$
2. $P(A) = P(A \wedge B) + P(A \wedge \neg B)$
3. $\Sigma_a P(A = a) = 1$: sum for all a in the value domain of A

Joint probability

Chim (C)	Non (N)	Bay (B)	P
T	T	T	0.0
T	T	F	0.2
T	F	T	0.04
T	F	F	0.01
F	T	T	0.01
F	T	F	0.01
F	F	T	0.23
F	F	F	0.5

Nếu có bảng xs đầy đủ → có thể tính ∀ xs luan.

Conditional probability

$$\boxed{P(A|B) = \frac{P(A,B)}{P(B)}}$$

- Properties of conditional probability
  - $P(A,B) = P(A|B)P(B)$
  - Chain rule: $P(A,B,C,D) = P(A|B,C,D)\,P(B|C,D)\,P(C|D)\,P(D)$
  - Conditional chain rule: $P(A,B|C) = P(A|B,C)\,P(B|C)$
  - Bayes' theorem $P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$
  - Conditional Bayes' theorem: $P(A|B,C) = \frac{P(B|A,C)\,P(A|C)}{P(B|C)}$
  - $P(A) = \Sigma_b\{P(A|B=b)\,P(B=b)\}$, sum for all values b of B
  - $P(\neg B|A) = 1 - P(B|A)$

Probabilistic independence

- An event A is said to be independent of another event B if $P(A|B) = P(A)$
  - Meaning: knowing that B occurred does not change the probability that A occurred
  - Therefore $P(A,B) = P(A)P(B)$

- A is conditionally independent of B given C if
  - $P(A|B,C) = P(A|C)$ or $P(B|A,C) = P(B|C)$
  - Meaning: B doesn't tell us anything about A if we already know C
  - Therefore $P(A,B|C) = P(A|C)P(B|C)$

Bayes' theorem

$$\boxed{P(A|B) = \frac{P(A).\,P(B|A)}{P(B)}}$$

ex

- A person has a positive result with disease B
- Testing equipment is not completely accurate
  - The equipment gives a positive result for 98% sick people
  - The equipment gives a positive result for 3% people who are not sick
- 0.8% of the population has this disease
- Question: Is this person sick?

- Notation: event that a person has disease is B, event that a person has a positive result is A
- According to the problem data we have:
  - $P(B) = 0.008, P(\neg B) = 1 - 0.008 = 0.992$
  - $P(A|B) = 0.98, P(\neg A|B) = 1 - 0.98 = 0.02$
  - $P(A|\neg B) = 0.03, P(\neg A|\neg B) = 1 - 0.03 = 0.97$
- We need to compare probabilities $P(B|A)$ and $P(\neg B|A)$
- Using Bayes' theorem:
  - $P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{0.98*0.008}{P(A)} = \frac{0.00784}{P(A)}$
  - $P(\neg B|A) = \frac{P(A|\neg B)P(\neg B)}{P(A)} = \frac{0.03*0.992}{P(A)} = \frac{0.02976}{P(A)}$
- $P(\neg B|A) > P(B|A)$, ➔ Not sick

- Compute $P(A|B,C)$, where B and C are conditionally independent given A
  - Bayes' theorem $P(A|B,C) = \frac{P(B,C|A)*P(A)}{P(B,C)}$
  - Probabilistic independence $P(B,C|A) = P(B|A) * P(C|A)$
  - Therefore $P(A|B,C) = \frac{P(B|A)*P(C|A)*P(A)}{P(B,C)}$

- Example:
  - Give 3 binary valuables: liver disease BG (A), jaundice VD (B), anemia TM (C)
  - Assume that VD is independent with TM
  - Know that $P(BG) = 10^{-7}$ $= P(A)$
  - Someone has VD disease
  - Know that $P(VD) = 2^{-10}$ and $P(VD|BG) = 2^{-3}$
  - a) What is the probability that a tester has disease?
  - b) Know that a person has TM disease and $P(TM) = 2^{-6}$, $P(TM|BG) = 2^{-1}$. Compute the probability that the tester has BG.

$P(B) = 2^{-10}$

$P(B|A) = 2^{-3}$

$P(C) = 2^{-6}$

$P(C|A) = 2^{-1}$

a) $P(A|B) = \dfrac{P(A).P(B|A)}{P(B)} = \dfrac{10^{-7}.2^{-3}}{2^{-10}} = \dfrac{10^{-7}}{2^{7}} = 5^{-7}$

b) $P(A|B,C) = \dfrac{P(B|A).P(C|A).P(A)}{P(B,C)} = \dfrac{2^{-3}.2^{-1}.10^{-7}}{2^{-10}.2^{-6}} = \dfrac{10^{-7}}{2^{-12}}$

Bayesian network

Example:

- Problem: A person comes home from work, need to guess if there is someone in the house?
- Know that:
  - If family members get away, the yard lights are often (not always) turned on
  - When there is no one at home, a dog is tied outside
  - If being sick, the dog is also tied outside
  - If the dog is outside, family members can hear the barking

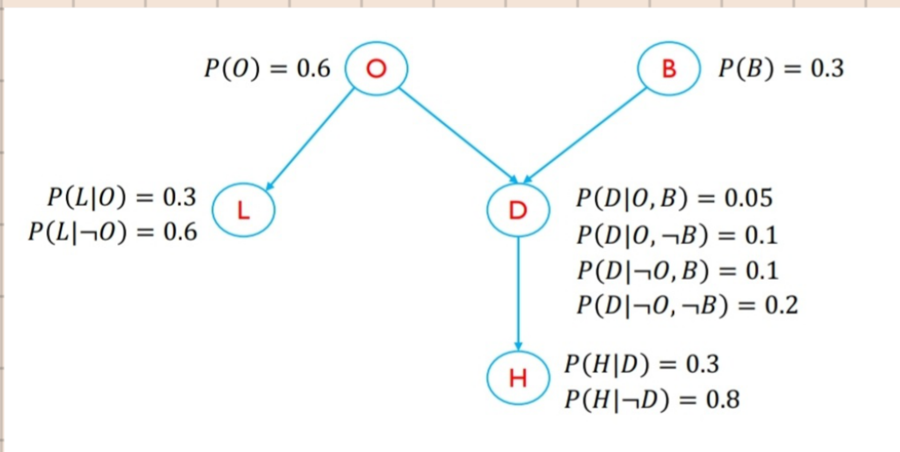
O: no one at home

L: yard lights is turned on

D: dog is tied outside

B: dog is sick

H: can hear the barking



Definition of Bayesian network

A Bayesian network includes 2 parts:

1, Directed acyclic graph : có hg, o chu trình

2, Conditional probability table.

ex1

$$
\begin{aligned}
P(H, \neg L, D, \neg O, B) &= P(H \mid \neg L, D, \neg O, B) . P(\neg L, D, \neg O, B) \\
&= P(H \mid D) . P(\neg L \mid D, \neg O, B) . P(D, \neg O, B) \\
&= P(H \mid D) . P(\neg L \mid \neg O) . P(D \mid \neg O, B) . P(\neg O, B) \\
&= P(H \mid D) . P(\neg L \mid \neg O) . P(D \mid \neg O, B) . P(\neg O) . P(B) \\
&= 0{,}3 . (1-0{,}6) . 0{,}1 . (1-0{,}6) . 0{,}3 \\
&= 0{,}00144
\end{aligned}
$$

Cách 2: $P(H, \neg L, D, \neg O, B) = P(H \mid D) . P(\neg L \mid \neg O) . P(D \mid \neg O, B) . P(\neg O) P(B)$

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i \mid parents(X_i))$$

ex2

nhà có ng̃, chó ốm, buộc ngoài sân, đèn o sáng, chó sủa.

$$
\begin{aligned}
P(\neg O, B, D, \neg L, H) &= P(\neg O) . P(B) . P(D \mid \neg O, B) . P(\neg L \mid \neg O) . P(H \mid D) \\
&= 0{,}00144
\end{aligned}
$$

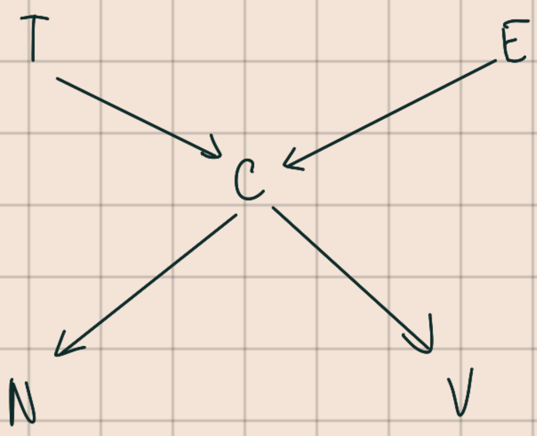
Bayesian Network Construction.

1. Define the set related random variables
2. Choose the order for variables
   Example: $X_1, X_2, \ldots, X_n$
3. **for** $i = 1$ **to** n **do**
   a. Add a node for X_i
   b. Select $parents(X_i)$ is the smallest set of given nodes so that X_i is conditionally independent of all previous nodes if knowing $parents(X_i)$
   c. Add a directed arc from each node $parents(X_i)$ to X_i
   d. Add conditional probability values $P(X_i|parents(X_i))$ or $P(X_i)$ if $parents(X_i) = \emptyset$

ex

- A person installed an anti-theft alarm system at home
- The system will alarm when there is a thief
- But, the system can alarm (inaccuratly) if there is a tremor by an earthquake
- In case hearing the alarm, 2 neighbors Nam and Việt will call the house onwer
- Due to many difference reasons, Nam and Việt can announce inaccurately, for instance, due to noise they can not hear any alarm or vice verser, they mistakes another sound for the alarm

T: having thief
E: earthquake
C: system alarm
N: Nam calls
V: Việt calls

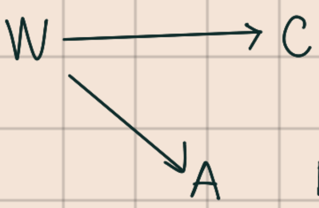


Bài tập xác suất 2

Giả sử cần suy diễn về quan hệ giữa thời tiết và giao thông. Cho ba biến ngẫu nhiên W, A, C biểu diễn cho ba tình huống sau: "thời tiết xấu" (*W*), "Chuyến bay Hà nội – HCM bị chậm" (*A*), "Quốc lộ 1 bị tắc" (*C*). Tiếp theo, giả sử chuyến bay chậm và đường tắc không ảnh hưởng đến nhau trong bất cứ thời tiết nào. Quan sát cho thấy, khi thời tiết xấu có 80% chuyến bay bị chậm và khi thời tiết tốt chỉ có 40% bị chậm. Tương tự, tần suất tắc quốc lộ 1 khi thời tiết xấu là 30% và khi thời tiết tốt là 10%. Xác suất thời tiết xấu tại Việt nam là 20%.

a) Vẽ mạng Bayes và bảng xác suất điều kiện cho ví dụ này.
b) Tính xác suất $P(\neg A, W, C)$;
c) Tính xác suất $P(A|C)$

$P(W) = 0{,}2$



$P(C|W) = 0{,}3$
$P(C|\neg W) = 0{,}1$

$P(A|W) = 0{,}8$
$P(A|\neg W) = 0{,}4$

$$\frac{P(A,W)}{P(W)} = P(A|W)$$

$$P(W|A) = \frac{P(A,W)}{P(A)}$$

b) $P(\neg A, W, C) = P(\neg A|W) . P(W) . P(C|W)$
$= (1 - 0{,}8) . 0{,}2 . 0{,}3$
$= 0{,}2 . 0{,}2 . 0{,}3$

c) $P(A|C) = P(A)$?

$$
\begin{aligned}
P(A,W) &= P(A,W,C) + P(A,W,\neg C) \\
&= P(A|W) \cdot P(W) \cdot P(C|W) + P(A|W) \cdot P(W) \cdot P(\neg C|W) \\
&= 0{,}8 \cdot 0{,}2 \cdot 0{,}3 + 0{,}8 \cdot 0{,}2 \cdot 0{,}7 \\
&= 0{,}16
\end{aligned}
$$

$$
\begin{aligned}
P(A,C) &= P(A,C,B) + P(A,C,\neg B) \\
&= P(A|B) \cdot P(C|B) \cdot P(B) + P(A|\neg B) \cdot P(C|\neg B) \cdot P(\neg B) \\
&= 0{,}4 \cdot 0{,}3 \cdot 0{,}6 + 0{,}7 \cdot 0{,}25 \cdot 0{,}4 \\
&= 0{,}072 + 0{,}07 \\
&= 0{,}142
\end{aligned}
$$

$$
\begin{aligned}
P(A) &= P(A,B) + P(A,\neg B) \\
&= P(A|B) \cdot P(B) + P(A|\neg B) \cdot P(\neg B) \\
&= 0{,}4 \cdot 0{,}6 + 0{,}7 \cdot 0{,}4 \\
&= 0{,}52
\end{aligned}
$$

$$
\begin{aligned}
P(C) &= P(C,B) + P(C,\neg B) \\
&= P(C|B) \cdot P(B) + P(C|\neg B) \cdot P(\neg B) \\
&= 0{,}3 \cdot 0{,}6 + 0{,}25 \cdot 0{,}4 \\
&= 0{,}28
\end{aligned}
$$

$P(A) \cdot P(C) = 0{,}1456 \neq P(A,C)$

$\rightarrow$ A, C is dependent

b)

$$
\begin{aligned}
P(C) &= P(C,A) + P(C,\neg A) \\
&= P(C|A) \cdot P(A) + P(C|\neg A) \cdot P(\neg A) \\
&= 0{,}3 \cdot 0{,}6 + 0{,}3 \cdot 0{,}4 \\
&= 0{,}3
\end{aligned}
$$

$P(C|A) = P(C) = 0{,}3$

$\rightarrow$ A, C is independent

Generally independent feature of probability: Definition of d-seperation (2/5)

- Definition *d-seperation* answer the question about the independence of set of nodes X with set of nodes Y when knowing the set of nodes E on a Bayes network
  - Nodes X and nodes Y are called as being *d-seperated* by nodes E if X and Y are independent when knowing E
  - Nodes X and nodes Y are *d-connection* if they are not *d-seperated*

- To define *d-seperation* of sets X and Y, we first define *d-seperation* between 2 single nodes x of X and y of Y
  - 2 sets of nodes will be independent if each node in one set in independent of all nodes in the other

- **Principle 1**: Node x and y are *d-connected* if there is an unblocked path *between 2 nodes*. In contrast, if there is no such path, x and y are *d-seperated*
  - A path is a sequence of contiguous arcs, regardless of the direction of the arcs
  - An unblocked path is a path on which no 2 adjacent arcs are directed at each other
  - Nút có hai cung hướng vào như vậy gọi là nút *xung đột*

$$x \longrightarrow r \longrightarrow s \longrightarrow t \longleftarrow v \longrightarrow y$$

→ nút xung đột

x → y: đg đi phong toả

x, y : d-separation

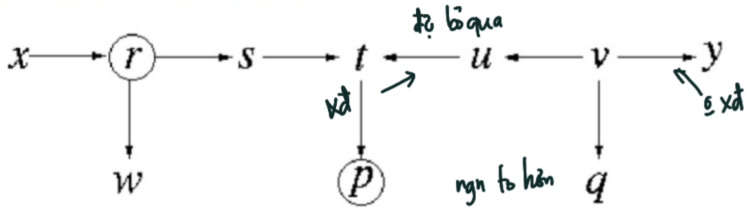
  - The connection and seperation features following **Principle 1** is unconditional and so the independence of probability is defined by **Principle 1** is unconditionally independent.

- **Principle 2**: node x and y are *d-conditional connected* when knowing the set of nodes E if there exists unblocked path (not include any confict node) and does not pass any node of E. In contrast, if there is no such path, we say that x and y are *d-seperated* by E. In other words, every paths between x and y (if any) are blocked by E.
  - When knowing the value of some nodes (set of nodes E), the independence or dependence between remanining nodes can be changed
  - the independence or dependence in this case is called as *d-conditional seperation* by the set of variables E

$$x \longrightarrow r \longrightarrow s \longrightarrow t \longleftarrow u \longleftarrow v \longrightarrow y$$

trên đg đc có nút ∈ tập E → phong toả

- **Principle 3**: If a conflict node is member of set E, or having descendant in set E, so that node does not block paths through it
  - Assume that we know an event is caused by 2 or more causes, if we already know 1 cause is true then the probability of the other causes is reduced, if we know 1 cause is false then the probability of other causes increases



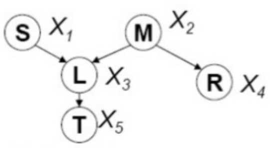
s and y are d-connection, x and u are d-seperation

Inference in Bayesian network

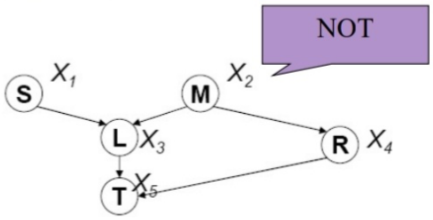
$$P(L \mid B, \neg H) = \frac{P(L, B, \neg H)}{P(B, \neg H)} = \frac{P(L, B, \neg H)}{P(B, \neg H, D) + P(B, \neg H, \neg D)}$$

Inference in reality

- Inference for a particular case
  - When network is in form of single connection (poly tree): there is no more than 1 path between any 2 nodes

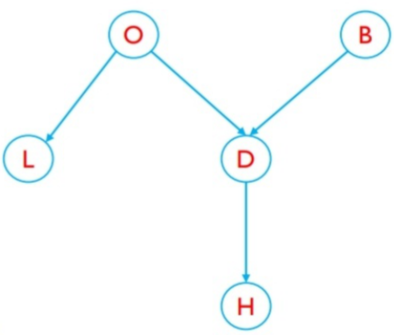


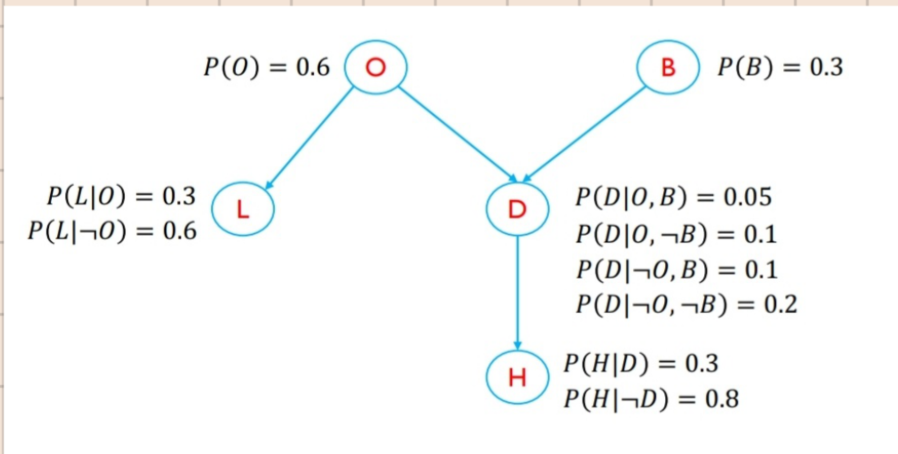
A poly tree



  - There exists an algorithm with linear complexity for poly tree
- Approximate inference by sampling

- The simplest case:
  - When the evidence E and result Q have only one direct connection
  - Distinguish 2 cases:
    - Causal inference (top to bottom): need to compute $P(Q|E)$ when E is the parent node of Q — suy diễn nhân quả
    - Diagnostic (bottom to top): need to compute $P(Q|E)$ when E is the child node of Q — suy diễn chẩn đoán





ex :

$$P(D|B) = \frac{P(D,B)}{P(B)} = \frac{P(D,B,O) + P(D,B,\neg O)}{P(B)}$$

$$= \frac{P(D|O,B) \cdot P(B) \cdot P(O) + P(D|O,\neg B) \cdot P(B) \cdot P(\neg O)}{P(B)}$$

$$= \frac{0{,}05 \cdot 0{,}3 \cdot 0{,}6 + 0{,}1 \cdot 0{,}3 \cdot 0{,}4}{0{,}3} = 0{,}07$$

ex

$$P(\neg B, \neg D) = \frac{P(\neg B, \neg D)}{P(D)} = \frac{P(\neg B, \neg D, O) + P(\neg B, \neg D, \neg O)}{P(\neg D)}$$

$$= \frac{P(\neg D|\neg B, O) \cdot P(\neg B) \cdot P(O) + P(\neg D|\neg B, \neg O) \cdot P(\neg B) \cdot P(\neg O)}{P(\neg D)}$$

$$= \frac{0{,}9 \cdot 0{,}7 \cdot 0{,}6 + 0{,}8 \cdot 0{,}7 \cdot 0{,}4}{P(\neg D)} = \frac{0{,}602}{P(\neg D)}$$

$$P(B|\neg D) = \frac{P(\neg D|B) \cdot P(B)}{P(\neg D)} = \frac{(1-0{,}07) \cdot 0{,}3}{P(\neg D)} = \frac{0{,}279}{P(\neg D)}$$

$$P(B|\neg D) + P(\neg B|\neg D) = \frac{0{,}279}{P(\neg D)} + \frac{0{,}602}{P(\neg D)} = 1$$

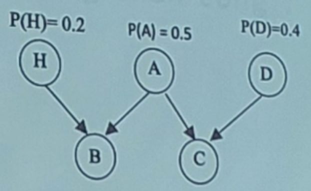
$$\frac{0{,}881}{P(\neg D)} = 1 \rightarrow P(\neg D) = 0{,}881$$

$$P(\neg B|\neg D) = \frac{0{,}602}{0{,}881} = 0{,}683.$$

General method

- Applies to both causal inference and diagnostic inference
  - **Step 1**: Convert conditional probability to simultaneous probability
  - **Step 2**: Use the independent feature of probability in Bayes network, rewrite simultaneous probability in form conditional probabilities of child node when knowing values of parent nodes
  - **Step 3**: Use probability values from conditional probability table to compute

Cho mạng Bayes sau, các biến có thể nhận giá trị {T,F} ({true, false})



H	A	P(B=T|A,H)
F	F	0.7
F	T	0.2
T	F	0.1
T	T	0.5

A	D	P(C=T|A,D)
F	F	0.8
F	T	0.3
T	F	0.4
T	T	0.2

a) Tính xác suất cả năm biến cùng nhận giá trị F.

b) Tính P(A|C).

c) Theo mạng đã cho H và B có độc lập xác suất với nhau không ?

a) $P(\neg H, \neg A, \neg D, \neg B, \neg C)$

$= P(\neg H) . P(\neg A) . P(\neg D) . P(\neg B \mid \neg H, \neg A) . P(\neg C \mid \neg A, \neg D)$

$= 0{,}8 . 0{,}5 . 0{,}6 . 0{,}3 . 0{,}2 = 0{,}0144$

b) $P(A \mid C) = \dfrac{P(A,C)}{P(C)} = \dfrac{P(A,C,D) + P(A,C,\neg D)}{P(C)}$

$= \dfrac{P(C \mid A,D) . P(A) . P(D) + P(C \mid A, \neg D) . P(A) . P(\neg D)}{P(C)}$

$= \dfrac{0{,}2 . 0{,}5 . 0{,}4 + 0{,}4 . 0{,}5 . 0{,}6}{P(C)} =$

$P(C) = P(A,C,D) + P(A,C,\neg D) + P(\neg A, C, D) + P(\neg A, C, \neg D)$

$= P(A) . P(D) . P(C \mid A, D) + \dots$

$0{,}5 . 0{,}4 . 0{,}2 + 0{,}5 . 0{,}6 . 0{,}4 + 0{,}5 . 0{,}4 . 0{,}3 + 0{,}5 . 0{,}6 . 0{,}8$

$= 0{,}49$

c) $P(H|B) = \dfrac{P(H,B)}{P(B)} = \dfrac{P(H,B,A) + P(H,B,\neg A)}{P(B)}$

$= \dfrac{0{,}2 \cdot 0{,}5 \cdot 0{,}5 + 0{,}2 \cdot 0{,}5 \cdot 0{,}1}{P(B)} = \dfrac{0{,}06}{P(B)}$

$P(B) = P(H,A,B) + P(H,B,\neg A) + P(\neg H,B,A) + P(\neg H,B,\neg A)$

$= 0{,}2 \cdot 0{,}5 \cdot 0{,}5 + 0{,}2 \cdot 0{,}5 \cdot 0{,}1 + 0{,}8 \cdot 0{,}5 \cdot 0{,}2 + 0{,}8 \cdot 0{,}5 \cdot 0{,}7$

$= 0{,}42$

$\Rightarrow P(H|B) = \dfrac{0{,}06}{0{,}42} = \dfrac{1}{7} \neq P(H)$

$\Rightarrow$ B and H are dependent

Machine learning

Decision Tree

: học từ quá trình đào tạo

- Is a tree-like classification model
  - Each mid-node (not a leaf node) corresponds to an features testing, each branch of node corresponding to a feature value at that node
  - Each leaf node corresponds to a label
- Classification process:
  - Sample goes from the root down to the bottom.
  - At each mid-node, features vector of that node is tested. Depending on feature value, sample moves down the corresponding branch
  - When reaching the leaf nod, sample is given a classification label

ID3 Algorithm

- Algorithm
  - **Init**: the current node is the root-node containing all of training data set
  - At the current node n, select features:
    - Unused at ancestor-node (previous node)
    - Allows to divide training data set into subsets **in the best way**
    - For each feature value selected, add a child-node below
    - Divide samples of current node into child-node by selected feature value
  - **Repeat** (recursively) until:
    - All features were used at above nodes, or
    - All samples of current node have the same label
    - Label of node is takend by the majority of labels of samples of current node

How to choose feature at each node?

Criteria for feature selection ID3.

- At each node n
  - The set (subset) of data corresponding to that node
  - Need to select the feature that allows the best splitting of the data set
- Criteria:
  - Data after being divided is as large as possible
  - Measure Information Gain - IG
  - **Select the feature with the largest IG**
  - IG is calculated based on entropy of set (subset) of data

$S = \{x_1, x_2, \dots x_n\}$

Node

$\uparrow$

$\max\{IG_1, IG_2, IG_3 \dots IG_n\}$

$\uparrow$

Entropy (S) : $H(S)$ → tính theo label

2 labels

$[3^+, 2^-] \rightarrow H(S) \approx 0,97$

$[n^+, n^-] \rightarrow H(S) = 1$

$[n^+, 0^-] \rightarrow H(S) = 0$

$[3^+, 4^-] \rightarrow H(S) = 0,985$

$[3^+, 1^-] \rightarrow H(S) = 0,811$

$[4^+, 2^-] \rightarrow H(S) = 0,918$

$[6^+, 1^-] \rightarrow H(S) = 0,592$

Entropy

$$\boxed{H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i}$$

Information Gain

$$\boxed{IG(S,A) = H(S) - \sum_{v \in A} \frac{|S_v|}{|S|} \cdot H(S_v)}$$

ex

- Trời = { mưa, nắng, u ám }

$S_{mưa} = [3^+, 2^-] \approx 0,971 = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\cdot\log_2\frac{2}{5}$

$S_{nắng} = [2^+, 3^-] \approx 0,971$

$S_{u\ ám} = [4^+, 0] \approx 0$

$$IG(S, \text{Trời}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u)$$

$$= 0{,}94 - \left(\frac{5}{14} \cdot 0{,}971 + \frac{5}{14} \cdot 0{,}971 + \frac{4}{14} \cdot 0 \right)$$

$$= 0{,}246$$

- Nhiệt độ = { nóng , TB, lạnh }

$$S_{\text{nóng}} = [2^+, 2^-] = 1$$

$$S_{\text{lạnh}} = [3^+, 1^-] = -\frac{3}{4} \cdot \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0{,}811$$

$$S_{TB} = [4^+, 2^-] = -\frac{4}{6} \cdot \log_2 \frac{4}{6} - \frac{2}{6} \cdot \log \frac{2}{6} = 0{,}918$$

$$IG(S, \text{Nhiệt độ}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u)$$

$$= 0{,}94 - \left(\frac{4}{14} \cdot 1 + \frac{4}{14} \cdot 0{,}811 + \frac{6}{14} \cdot 0{,}918 \right)$$

$$= 0{,}029$$

- Độ ẩm = { cao , TB }

$$S_{\text{cao}} = [3^+, 4^-] = 0{,}985$$

$$S_{TB} = [6^+, 1^-] = 0{,}592$$

$$IG(S, \text{Độ ẩm}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u)$$

$$= 0{,}94 - \left(\frac{7}{14} \cdot 0{,}985 + \frac{7}{14} \cdot 0{,}592 \right)$$

$$= 0{,}1515$$

Calculate $IG(S, Gió)$

$$values(Gió) = \{yếu, mạnh\}$$

$$S = [9+, 5-], H(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$S_{yếu} = [6+, 2-], \mathrm{H}(S_{yếu}) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.811$$

$$S_{mạnh} = [3+, 3-], \mathrm{H}(S_{mạnh}) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$$

$$IG(S, Gió) = H(S) - \frac{8}{14}\mathrm{H}(S_{yếu}) - \frac{6}{14}\mathrm{H}(S_{mạnh})$$
$$= 0.94 - \frac{8}{14}0.811 - \frac{6}{14}1$$
$$= 0.048$$

ex

Bài tập 1

Cho tập dữ liệu như bảng bên dưới, trong đó Chiều cao, Dân tộc, Tính tình là các thuộc tính, f là nhãn phân loại.

Số TT	f	Chiều cao	Dân tộc	Tính tình
1	+	Cao	Lào	Vui vẻ
2	+	Thấp	Việt	Vui vẻ
3	-	Cao	Việt	Ôn hoà
4	-	Thấp	Thái	Vui vẻ
5	-	Cao	Thái	Vui vẻ
6	-	Cao	Lào	Ôn hòa

a) Xây dựng cây quyết định sử dụng thuật toán ID3.

b) Sử dụng phương pháp phân loại Bayes đơn giản tìm nhãn cho mẫu (**Chiều cao = "Thấp", Dân tộc = "Việt", Tính tình = "Ôn hoà"**).

a) $H(S) = -2\log_2\frac{4}{6} - 4\log_2\frac{2}{6}$
$= 0,918$

* Chiều cao = { cao, thấp }

$S_{cao} = [1^+, 3^-] = 0,811$

$S_{thấp} = [1^+, 1^-] = 1$

$$IG(S, \text{Chiều cao}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u) = 0,918 - \frac{4}{6} \cdot 0,811 - \frac{2}{6} \cdot 1$$
$$= 0,044$$

* Dân tộc = { Lào, Việt, Thái }

$S_L = [1^+, 1^-] = 1$

$S_V = [1^+, 1^-] = 1$

$S_T = [0^+, 2^-] = 0$

$$IG(S, \text{Dân tộc}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u) = 0,918 - \frac{2}{6} \cdot 1 - \frac{2}{6} \cdot 1 - \frac{2}{6} \cdot 0 = 0,251$$

* Tính tình = { Vui vẻ, ôn hoà }

$S_{vui\ vẻ} = [2^+, 2^-] = 1$

$S_{ôn\ hoà} = [0^+, 2^-] = 0$

$$IG(S, \text{Tính tình}) = H(S) - \sum_{u \in A} \frac{|S_u|}{|S|} \cdot H(S_u) = 0,918 - \frac{4}{6} \cdot 1 - \frac{2}{6} \cdot 0 = 0,251$$

b)

$$y = \operatorname{argmax}_{c \in \{Y, N\}} P(\text{thấp} \mid c) \cdot P(\text{Việt} \mid c) \cdot P(\text{ôn hoà} \mid c) \cdot P(c)$$

$$= \operatorname{argmax} \begin{pmatrix} P_1 = P(\text{thấp} \mid Y) \cdot P(\text{Việt} \mid Y) \cdot P(\text{ôn hoà} \mid Y) \cdot P(Y) \\ P_2 = P(\text{thấp} \mid N) \cdot P(\text{Việt} \mid N) \cdot P(\text{ôn hoà} \mid N) \cdot P(N) \end{pmatrix}$$

$P(Y) = \frac{2}{6}$ $\quad$ $P(N) = \frac{4}{6}$

$P(\text{thấp} \mid Y) = \frac{1}{2}$ $\quad$ $P(\text{Việt} \mid Y) = \frac{1}{2}$ $\quad$ $P(\text{ôn hoà} \mid Y) = \frac{0}{2}$

$P(\text{thấp} \mid N) = \frac{1}{2}$ $\quad$ $P(\text{Việt} \mid No) = \frac{1}{2}$ $\quad$ $P(\text{ôn hoà} \mid N) = \frac{2}{2}$.

$P_1 = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{0}{2} \cdot \frac{2}{6} = 0$

$P_2 = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{2}{2} \cdot \frac{4}{6} = \frac{1}{6}$

$\max(P_1, P_2) = \frac{1}{6} \Rightarrow y = N$

Naive Bayes Classification

- In training phase, we a set of samples, each sample is a pair $< \boldsymbol{x_i}, y_i >$, where
  - $\boldsymbol{x_i}$ features vector
  - y_i is label, $y_i \in C$ (C is set of labels)
- After training, classifier need predict label y for new sample $\boldsymbol{x} =< x_1, x_2, \dots, x_n >$

$$y = argmax_{c_j \in C} P(c_j | x_1, x_2, \dots, x_n)$$

- Using Bayes principle:

$$y = argmax_{c_j \in C} \frac{P(x_1, x_2, \dots, x_n | c_j) P(c_j)}{P(x_1, x_2, \dots, x_n)}$$

$$= argmax_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j)$$

Naive Bayes classification (2/2)

Frequency of observing the label c_j on dataset D: $\frac{count(c_j)}{|D|}$

$$y = argmax_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j)$$

Using theory about independence probability (**Đơn giản!!!**)

$$P(x_1, x_2, \dots, x_n | c_j) = P(x_1 | c_j) P(x_2 | c_j) \dots P(x_n | c_j)$$

Number of occurrence x_i with c_j divided by number of occurence c_j: $\frac{count(x_i, c_j)}{count(c_j)}$

ex

Ngày	Trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	nắng	nóng	cao	yếu	không
D2	nắng	nóng	cao	mạnh	không
D3	u ám	nóng	cao	yếu	có
D4	mưa	trung bình	cao	yếu	có
D5	mưa	lạnh	bình thường	yếu	có
D6	mưa	lạnh	bình thường	mạnh	không
D7	u ám	lạnh	bình thường	mạnh	có
D8	nắng	trung bình	cao	yếu	không
D9	nắng	lạnh	bình thường	yếu	có
D10	mưa	trung bình	bình thường	yếu	có
D11	nắng	trung bình	bình thường	mạnh	có
D12	u ám	trung bình	cao	mạnh	có
D13	u ám	nóng	bình thường	yếu	có
D14	mưa	trung bình	cao	mạnh	không

$$y = argmax_{c \in \{có, không\}} P(nắng | c) . P(TB | c) . P(cao | c) . P(mạnh | c) . P(c)$$

$$= argmax \begin{cases} p_1 = P(nắng | có) . P(TB | có) . P(cao | có) . P(mạnh | có) . P(có) \\ p_2 = P(nắng | k) . P(TB | k) . P(cao | k) . P(mạnh | k) . P(k) \end{cases}$$

$P(k) = \frac{5}{14}$ $\qquad P(có) = \frac{9}{14}$

$P(nắng | k) = \frac{3}{5}$ $\qquad P(TB | k) = \frac{2}{5}$ $\qquad P(cao | k) = \frac{4}{5}$ $\qquad P(mạnh | k) = \frac{3}{5}$

$P(nắng | có) = \frac{2}{9}$ $\qquad P(TB | có) = \frac{4}{9}$ $\qquad P(cao | có) = \frac{3}{9}$ $\qquad P(mạnh | có) = \frac{3}{9}$

$p_1 = \frac{2}{9} \cdot \frac{4}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} = 0{,}007$ $\qquad max = 0{,}041 \rightarrow y = k.$

$p_2 = \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} = 0{,}041$

Instance-based learning

- No building model
- Only save training samples
- Define a label for a new sample based on samples in data set that are similar to the new sample.
- Called as "lazy learning"

KNN (K-nearest neighbors algorithm)

- Assume that sample x has feature values
  $< a_1(x), a_2(x), \dots, an(x) >$, where $a_i(x)$ is real number.
- Distance between 2 samples x_i and x_j is the Euclidean distance:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{n} (a_l(x_i) - a_l(x_j))^2}$$

Algorithm

Learning phase (training)
Save training samples of the form <x, f(x)> into the database
Classification phase
Input: parameter k
For sample x to be classified:

1. Calculate the distance d(x, x_i) from x to all samples xi in the database
2. Find k samples with the smallest d(x, x_i), assuming those k samples are x_1, x_2, ..., x_k.
3. Determine the classification label f'(x) is the label that occupies the majority in the set $\{x_1, x_2, \dots, x_k\}$

Câu 4 (3 điểm)

Cho bảng dữ liệu huấn luyện dưới đây, các dòng A, B, C là thuộc tính, D là nhãn phân loại.

A	2	2	1	1	2	1	2	1
B	1	2	1	2	1	1	2	2
C	1	2	1	1	2	2	1	2
D	+	+	+	+	-	-	-	-

a) Sử dụng thuật toán k láng giềng (với k = 3) tìm nhãn phân loại cho mẫu sau:
A = 2, B = 2, C = 1.
chỉ rõ kết quả của theo từng bước tính toán.

b) Tìm nút gốc của cây quyết định sử dụng thuật toán ID3 cho dữ liệu trên.

Chú ý: Trong trường hợp có các thuộc tính với cùng độ ưu tiên thì chọn thuộc tính theo thứ tự bảng chữ cái.

a)

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
A	2	2	1	1	2	1	2	1	2
B	1	2	1	2	1	1	2	2	2
C	1	2	1	1	2	2	1	2	1
D	+	+	+	+	-	-	-	-	?

$$d(S_9, S_1) = \sqrt{(2-2)^2 + (1-2)^2 + (1-1)^2} = 1$$
$$d(S_9, S_2) = \sqrt{(2-2)^2 + (2-2)^2 + (2-1)^2} = 1$$
$$d(S_9, S_3) = \sqrt{(1-2)^2 + (1-2)^2 + (1-1)^2} = \sqrt{2}$$
$$d(S_9, S_4) = \sqrt{(1-2)^2 + (2-2)^2 + (1-1)^2} = 1$$
$$d(S_9, S_5) = \sqrt{(2-2)^2 + (1-2)^2 + (2-1)^2} = \sqrt{2}$$
$$d(S_9, S_6) = \sqrt{(1-2)^2 + (1-2)^2 + (2-1)^2} = \sqrt{3}$$
$$d(S_9, S_7) = \sqrt{(2-2)^2 + (2-2)^2 + (1-1)^2} = 0$$
$$d(S_9, S_8) = \sqrt{(1-2)^2 + (2-2)^2 + (2-1)^2} = \sqrt{2}$$

With $k = 3 \Rightarrow$ we have 3 samples S_1, S_2, S_7

$\Rightarrow f'(x) = +$

$\Rightarrow A = 2, B = 2, C = 1, D = +$

Bài tập Học máy

1. Cho dữ liệu huấn luyện như trong bảng (f là nhãn phân loại).

a) Hãy xây dựng cây quyết định sử dụng thuật toán ID3. Trong trường hợp có hai thuộc tính tốt tương đương thì chọn theo thứ tự bảng chữ cái.

b) Giả sử không biết nhãn phân loại của ví dụ cuối cùng, hãy xác định nhãn cho ví dụ đó bằng phương pháp Bayes đơn giản (chỉ rõ các xác suất điều kiện thành phần) và k láng giềng gần nhất với k = 5.

X	Y	Z	f
1	0	1	1
1	1	0	0
0	0	0	0
0	1	1	1
1	0	1	1
0	0	1	0
0	1	1	1
1	1	1	0

a) $S = [4^+, 4^-] \Rightarrow H(S) = 1$

* $X = \{0, 1\}$

$S_0 = [2^+, 2^-] \rightarrow 1$

$S_1 = [2^+, 2^-] \rightarrow 1$

$IG(S, X) = 1 - \frac{4}{8} \cdot 1 - \frac{4}{8} \cdot 1 = 0$

* $Y = \{0, 1\}$

$S_0 = [2^+, 2^-] \longrightarrow 1$

$S_1 = [2^+, 2^-] \rightarrow 1$

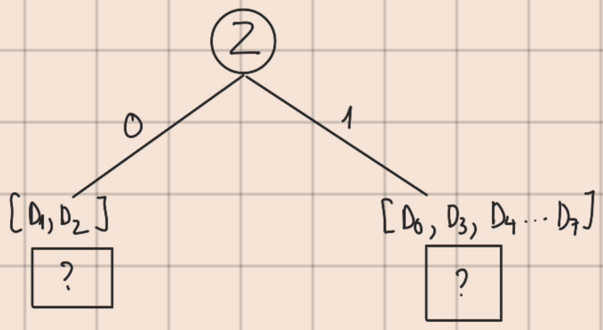
$IG(S, Y) = 1 - \frac{4}{8} \cdot 1 - \frac{4}{8} \cdot 1 = 0$

* $Z = \{0, 1\}$

$S_0 = [0^+, 2^-] \rightarrow 0$

$S_1 = [4^+, 2^-] \rightarrow 0{,}918$

$IG(S, Z) = 1 - \frac{2}{8} \cdot 0 - \frac{6}{8} \cdot 0, = 0{,}3115$



$\Rightarrow$ Z is the root node.

* $[D_1, D_2]$: $S_1 = [0^+, 2^-] \rightarrow 0$

* $X = \{0, 1\}$, $S_0 = [0^+, 1^-] \rightarrow 0$

$S_1 = [0^+, 1^-] \rightarrow 0$

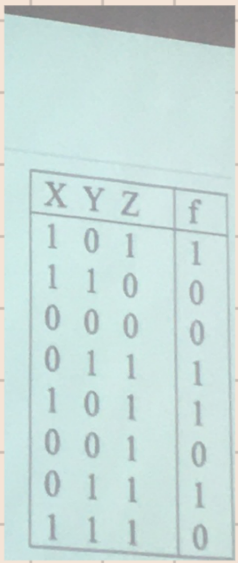
$IG(S_1, X) = 0$

* $Y = \{0, 1\}$ $S_0 = [0^+, 1^-] \rightarrow 0$

$S_1 = [0^+, 1^-] \rightarrow 0$

$IG(S_1, Y) = 0$

$* \; [D_0, D_2, D_3, \dots D_7]$ $\qquad S_2 = [4^+, 2^-] \Rightarrow 0{,}918$



X	Y	Z	f
1	0	1	1
1	1	0	0
0	0	0	0
0	1	1	1
1	0	1	1
0	0	1	0
0	1	1	1
1	1	1	0

$* \; X = \{0, 1\}$ $\qquad S_0 = [2^+, 1^-] \Rightarrow 0{,}918$

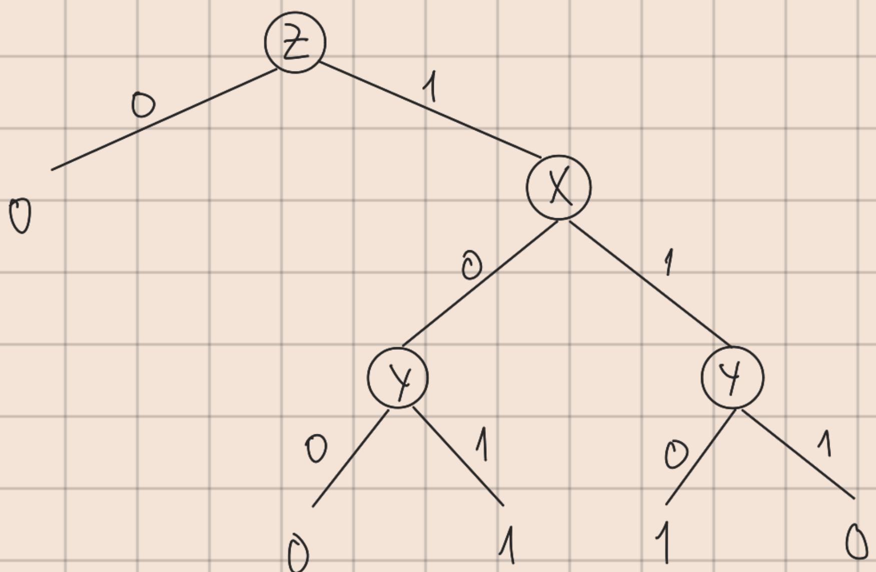
$\qquad S_1 = [2^+, 1^-] \Rightarrow 0{,}918$

$$IG(S_2, X) = 0{,}918 - \frac{3}{6} \cdot 0{,}918 - \frac{3}{6} 0{,}918 = 0$$

$* \; Y = \{0, 1\}$ $\qquad S_0 = [2^+, 1^-] \Rightarrow 0{,}918$

$\qquad S_1 = [2^+, 1^-] \Rightarrow 0{,}918$

$$IG(S_2, Y) = 0$$



b) $y = \operatorname{argmax}_{c \in \{0,1\}} \left(P(1|c) . P(1|c) . P(1|c) . P(c) \right)$

$$= \operatorname{argmax} \begin{pmatrix} P_3 = P(1|1) . P(1|1) . P(1|1) . P(1) \\ P_4 = P(1|0) . P(1|0) . P(1|0) . P(0) \end{pmatrix}$$

$P_0 = \frac{3}{7}$ $\qquad P_1 = \frac{4}{7}$.

$P_X(1|1) = \frac{2}{4}$ $\qquad P_X(1|0) = \frac{1}{3}$ $\qquad P_3 = \frac{\cancel{2}}{\cancel{4}} \cdot \frac{\cancel{2}}{\cancel{4}} \cdot \frac{\cancel{4}}{\cancel{4}} \cdot \frac{4}{7} = \frac{1}{7}$

$P_Y(1|1) = \frac{2}{4}$ $\qquad P_Y(1|0) = \frac{1}{3}$

$P_Z(1|1) = \frac{4}{4}$ $\qquad P_Z(1|0) = \frac{1}{3}$ $\qquad P_4 = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{\cancel{3}} \cdot \frac{\cancel{3}}{7} = \frac{1}{63}$.

$$\max\{P_3, P_4\} = \frac{1}{7} \; (P_3) \Rightarrow y = 1$$

KNN

$d(S_8, S_1) = 1$

$d(S_8, S_2) = 1$

$d(S_8, S_3) = \sqrt{3}$

$d(S_8, S_4) = 1$

$d(S_8, S_5) = 1$

$d(S_8, S_6) = \sqrt{2}$

$d(S_8, S_7) = 1$

$S_1, S_2, S_4, S_5, S_7 \Rightarrow 1$

Báo cáo

1. Intro
2. KB (CNN, LSTM, ...)
3. Program
   - Platform
   - Structure...
4. Result
5. Conclusion