TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

LOGIC ỨNG DỤNG TRONG TIN HỌC

GS.TSKH HOÀNG VĂN KIẾM

LOGIC
MỘT SỐ VẤN ĐỀ LÝ THUYẾT & ỨNG DỤNG TRONG TIN HỌC

KHÁI QUÁT VỀ LOGIC

- Aristote với thuyết "tam đoạn luận"
- Thế kỷ XVII, XVIII K.Leibnitz, G.Boole phát triển thành logic mệnh đề
- Chứng minh kiểu hình học Euclide (Toán)
- Thế kỷ XX khủng hoảng lớn, do chưa có lập luận đúng

KHÁI QUÁT VỀ LOGIC (tt)

- ***Trong khoa học, đời sống*** có nhiều loại logic
  – Danh từ: logic... ∀
  – Tính từ: tư duy logic, phát biểu logic
- ***Trong thực tế***: có logic trong ngôn ngữ → biểu diễn chính xác ngữ nghĩa của câu nói → khắc phục tính nhập nhằng
- ***Trong suy luận:*** logic không hình thức

KHÁI QUÁT VỀ LOGIC (tt)

- ***Logic trong khoa học***
  - Logic nằm ngoài nội dung, xử lý theo hình thức, mệnh đề, phép toán
  - Một câu, một phát biểu trong thực tế rút gọn thành một ký hiệu
    - Logic mệnh đề : mệnh đề → 1 ký hiệu
    - Dùng các ký hiệu: $\wedge$, $\vee$, $\neg$ ($\rightarrow$)
- **Leibnitz**: xây dựng số nhị phân và phát hiện ra sự trùng hợp giữa số nhị phân và kinh dịch
- **Boole**: logic mệnh đề

NHỮNG HẠN CHẾ CỦA LOGIC MỆNH ĐỀ

$$P \rightarrow Q \qquad \ldots \quad (*)$$

1) P đúng
Q đúng
} thì (*) đúng

1) (*) đúng
Q sai
} thì P sai

3) $P \rightarrow Q$
$Q \rightarrow R$
} thì $P \rightarrow R$

NHỮNG HẠN CHẾ CỦA LOGIC MỆNH ĐỀ (tt)

- Thường có những suy luận không chắc chắn (khi đúng, khi sai)

1) P đúng, Q đúng } thì $P \xrightarrow{\text{đúng}} Q$ ⇒ 1') P sai, $P \xrightarrow{\text{đúng}} Q$ } ? Q sai

2) $P \xrightarrow{\text{đúng}} Q$, Q sai } thì P sai ⇒ 2') Q sai, $P \xrightarrow{\text{đúng}} Q$ } ? P đúng

3) P → Q, Q → R } nên P→R ⇒ $P \xrightarrow{Q \sim Q'} R$?

NHỮNG HẠN CHẾ CỦA LOGIC MỆNH ĐỀ (tt)

- Những tiền đề logic vẫn còn có nhiều thắc mắc.

Ví dụ: p → q ≈ $\overline{p} \vee q$

p	q	p ∨ q	p ∧ q	p → q	$\overline{p} \vee q$
1	1	1	1	1	1
1	0	1	0	**0**	0
0	1	1	0	**1**	1
0	0	0	0	**1**	1

NHỮNG HẠN CHẾ CỦA LOGIC MỆNH ĐỀ (tt)

- Vì sao P, Q chỉ có giá trị 0, 1 → ***logic đa trị***
- P, Q có giá trị được tin cậy mức nào?

→ logic xác suất, logic khả suất

Ví dụ: về màu sắc; P màu xanh → ? xanh đậm, xanh nhạt ...

- Ký hiệu biểu diễn mức độ tin cậy $\xrightarrow{?}$

Ví dụ:

$$\left.\begin{array}{l} P \xrightarrow{1.7} Q \\ Q \xrightarrow{1.8} R \end{array}\right\} \quad P \xrightarrow{?} R$$

NHẬN XÉT

- Mức độ cao nhất của logic là logic mờ (fuzzy logic) . H.Zahde – 1970's
- *Ví dụ*:
  – Trời mưa to → mưa (trời, to)
  – Tôi ăn cơm → ăn(tôi, cơm)

LOGIC TÍNH TOÁN
VÀ GIẢI PHƯƠNG TRÌNH

SỐ LOGIC

- Số biểu diễn logic của một biến A có giá trị : 0 1
- Ký hiệu số logic của biến A là: #A

*** Xét trong không gian 1 biến: AB**

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

***Ma trận biểu diễn:** [A, B]*

*** Xét trong không gian 3 biến: ABC**

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

***Ma trận biểu diễn:** [A, B, C]*

SỐ LOGIC (tt)

*** Xét trong không gian n biến: ABC...**

$$\begin{pmatrix} 0 \\ 1 \\ \cdots \\ \cdots \\ \\ \cdots \\ 2^n - 1 \end{pmatrix}$$

Ma trận biểu diễn: $[A_1, A_2, \ldots A_n]$

SỐ LOGIC (tt)

- Dựa vào ma trận input → xây dựng một tập các giá trị f

A	B	C	f1	f1	f3	...	...	f_k
0	0	1	1	1	0	...	?	0
1	0	0	...	...	...			
...	...	...	...					
...	...							

- NHẬN XÉT:

$$\mathbf{f(A, B, C) \cong \# f(A, B, C)}$$

Một hàm có một số logic → một số logic ứng với một hàm

HỆ QUẢ

❶ Số logic của một biến hợp (một tổng) bằng tổng các logic thành phần

$$\# (A + B) = \# A + \# B$$

hay $$\# (A \vee B) = \# A \vee \# B$$

❷ Số logic của một tích bằng tích các logic thành phần

$$\# (A . B) = \# A . \# B$$

hay $$\# (A \wedge B) = \# A \wedge \# B$$

❸ $$\# \bar{A} = \overline{\# A}$$

HỆ QUẢ (tt)

Chứng minh biểu thức

$$(A \wedge B) \vee (C \wedge D) \vee (A \wedge B \vee C \wedge D) \cong (A \vee B)(C \wedge D)$$

- *Cách 1:* Chứng minh công thức trên 12 luật logic của Vương Hạo, 1962 → cách này không dễ
- *Cách 1:* Lập bảng chân lý → cách này phức tạp
- *Cách 3:* Dùng số logic

Chứng minh đẳng thức **f(A, B, C)** $\cong$ **g(A, B, C)** *tương đương với chứng minh đẳng thức* **#f(A, B, C) = #g(A, B, C);** mà tính một số logic thì rất dễ

HỆ QUẢ (tt)

Chứng minh biểu thức: $A \longrightarrow B \cong \bar{B} \longrightarrow \bar{A}$

Caùch 1: $A \vee B \cong \bar{\bar{B}} \vee \bar{A}$

$B \vee \bar{A}$

$\bar{A} \vee B$

Caùch 2:

A	B	A→B	$\bar{B} \longrightarrow \bar{A}$
1			
1			

Caùch 3: **Tính số** logic. **Xét t**rong không gian [A , B]

$\#(A \longrightarrow B) = \#(\bar{B} \longrightarrow \bar{A})$

$\#(\bar{A} \vee B) = \#(\bar{\bar{B}} \vee \bar{A})$

$\#(\bar{A} + B) = \#(\bar{\bar{B}} + \bar{A})$

HỆ QUẢ (tt)

- Khi có n biến → biểu diễn số gồm 2^n

A_1	0 1 0 1 0 1	...	0 1			xen kẽ 1 số 0
A_2	00 11 00 11	...	11			xen kẽ 2 số 0
A_3	0000 1111	...	1111			xen kẽ 4 số 0
A_4	00000000	...	11111111			xen kẽ 8 số 0
... ...						
A_n	$\underbrace{00 \quad \ldots \quad 00}_{2^{n-1}}$	...	$\underbrace{11 \quad \ldots \quad 111}_{2^{n-1}}$			

- Từ $A_1,\ldots,A_n$ tính số logic liên quan, biểu thức liên quan

Bài tập:

Input vào các biến, cho các hàm bất kỳ. Chứng minh rằng các hàm có tương hợp hay không?

TUYỂN CHUẨN – HỘI CHUẨN

- Chuẩn hoá một biểu diễn dưới dạng biểu thức logic

Biểu thức

$$f: A, B, C \rightarrow \overline{A \vee B} \vee C \wedge D$$

Tuyển chuẩn

$$\overline{A}BC \vee \overline{A}BC \vee \ldots$$

Hội chuẩn

$$(A + B + \overline{C}) \cdot (\overline{A} + B + C) \cdot \ldots$$

Tuyển chuẩn: Tổng các tích. Xét số **1** ở vị trí nào thì đưa tổ hợp tại vị trí đó vào biểu thức logic của f.

Hội chuẩn: Tích các tổng. Xét số 0 ở vị trí nào thì đưa tổ hợp tại vị trí đó vào biểu thức logic của f.

TUYEÅN CHUAÅN – HOÄI CHUAÅN (tt)

Trường hợp 3 biến → $2^3 = 8$ tổ hợp. Lập tổ hợp các biến theo qui tắc

$$\left\{\begin{array}{l} A \\ \overline{A} \end{array}\right. \qquad \left\{\begin{array}{l} B \\ B \\ \overline{B} \\ \overline{B} \end{array}\right. \qquad \left\{\begin{array}{l} \overline{C} \\ \overline{C} \\ \overline{C} \\ \overline{C} \\ C \\ C \\ C \\ C \end{array}\right.$$

TUYỂN CHUẨN

$\overline{A}$	$\overline{B}$	$\overline{C}$	1	0	0	0	0	0	0	0	**C1**
A	$\overline{B}$	$\overline{C}$	0	1	0	0	0	0	0	0	**C2**
$\overline{A}$	B	$\overline{C}$	0	0	1	0	0	0	0	0	**C3**
A	B	$\overline{C}$	0	0	0	1	0	0	0	0	**C4**
$\overline{A}$	$\overline{B}$	C	0	0	0	0	1	0	0	0	**C5**
A	$\overline{B}$	C	0	0	0	0	0	1	0	0	**C6**
$\overline{A}$	B	C	0	0	0	0	0	0	1	0	**C7**
A	B	C	0	0	0	0	0	0	0	1	**C8**

TUYỂN CHUẨN (tt)

VÍ DỤ:

Cho #f : 10101001

- *Số 1 tại vị trí nào thì lấy tổ hợp tại vị trí đó vào*
- #f có dạng tuyển chuẩn là:

C1 + C3 + C5 + C8

1000 0000 + 0010 0000 + 0000 1000 + 0000 0001

$$f = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C + ABC$$

TUYỂN CHUẨN (tt)

PHƯƠNG PHÁP

1. Tính #f

1. Xét biểu diễn #f

$$\# f: (i_1\ i_2 \dots\ i_2{}^n)$$

$$i_k = 1 \rightarrow f: = f + C_k$$

3. Nếu **$k \le 2^n$** quay lại bước 2

Ngược lại dừng

HỘI CHUẨN

$\overline{A}$	$\overline{B}$	$\overline{C}$	1	1	1	1	1	1	1	0	**D8**
A	$\overline{B}$	$\overline{C}$	1	1	1	1	1	1	0	1	**D7**
$\overline{A}$	B	$\overline{C}$	1	1	1	1	1	0	1	1	**D6**
A	B	$\overline{C}$	1	1	1	1	0	1	1	1	**D5**
$\overline{A}$	$\overline{B}$	C	1	1	1	0	1	1	1	1	**D4**
A	$\overline{B}$	C	1	1	0	1	1	1	1	1	**D3**
$\overline{A}$	B	C	1	0	1	1	1	1	1	1	**D2**
A	B	C	0	1	1	1	1	1	1	1	**D1**

HỘI CHUẨN (tt)

VÍ DỤ:

Cho #f : 1011 1111 + 110 1111 + 1111 1110

- *Số 0 tại vị trí nào thì lấy tổ hợp tại vị trí đó vào*
- #f có dạng tuyển chuẩn là:

D2 + D4 + D8

1011 1111 + 1110 1111 + 1111 1110

$$f = (A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$$

HỘI CHUẨN (tt)

PHƯƠNG PHÁP

1. Tính #f

1. Xét biểu diễn #f

$$\# f: (i_1\ i_2 \ldots\ i_{2^n})$$

$$i_k = 0 \rightarrow f := f + D_k$$

3. Nếu $k \leq 2^n$ quay lại bước 2

Ngược lại dừng

PHƯƠNG TRÌNH LOGIC VÀĐỘC LẬP LOGIC

PHƯƠNG TRÌNH LOGIC

$$\begin{cases} \#F_1\ (A_1, A_1, ..., A_n) \\ \#F_1\ (A_1, A_1, ..., A_n) \\ ... \\ ... \\ \#F_m\ (A_1, A_1, ..., A_n) \end{cases}$$

<u>Giả sử có:</u>

$$\left.\begin{array}{l} F_1: (A \rightarrow B) \wedge D \\ F_2: (AC \vee D) \rightarrow B \\ F_3: ABC \rightarrow D \end{array}\right\}$$

➔ *tìm quan hệ xem F_1 ... F_n độc lập hay phụ thuộc logic*

➔ *tìm phương trình quan hệ.*

ĐỘC LẬP LOGIC

f1, f2, ..., fn độc lập logic khi và chỉ khi:

$\exists F$: F(f1, f2, ..., fn) = I (hằng số)

* **ĐỊNH NGHĨA**

f1, f2, ..., fn độc lập logic

$\Leftrightarrow$ ma trận biểu diễn số logic của từng f_{i} : { #f1 #f2 .... #fn } là 2^n số 0,1 ... 2^n -1

$$\begin{array}{l} \#f1 \\ \#f2 \\ \ldots \quad \ldots \\ \ldots \quad \ldots \\ \#fn \end{array} \begin{pmatrix} 0 & 1 & 0 & \ldots. & 1 \\ 1 & 0 & 1 & \ldots. & \\ & & & & \\ & & & & \\ 1 & 0 & 1 & \ldots. & 1 \end{pmatrix}$$

ĐỘC LẬP LOGIC (tt)

f1: $AB \vee C \to D$

f2: $A \vee \overline{B} \wedge \overline{C} \vee D$

$\left.\begin{array}{l} f1 \to f2 \\ f1 \leftrightarrow f2 \\ f2 \to f1 \end{array}\right\}$ trường hợp riêng F(f1, f2) =I

f1: $AB + \overline{A}\,\overline{B}$ → #f1 1001

f2: $\overline{B}$ → #f2 1100

Ta có 4 cột khác nhau → có f1, f2 là độc lập logic

Với	#A:	0101	$\#\overline{A}$: 1010
	#B:	0011	$\#\overline{B}$: 1100
	#AB:	0001	$\#\overline{A}\,\overline{B}$: 1000

ĐỘC LẬP LOGIC (tt)

THUẬT TOÁN: Tìm số logic từng #fi → lập thành ma trận, xét tính độc lập logic

- Nếu độc lập thì dừng
- Ngược lại
  - Xác định các giá trị tạo thành từ các cột $C_{i1}, C_{i2}, \ldots, C_{im}$
  - Biểu diễn

    #F : [0 1 … 1 …] với
    $$\begin{cases} 1 & \text{ở các vị trí } C_{i1}, C_{i2}, \ldots, C_{im} \\ 0 & \text{ở các vị trí khác} \end{cases}$$
- Số #F cộng logic từng cột, nếu có một Cij = 1 thì bằng 1, ngược lại thì bằng 0
- Nếu $C_{i1} \neq C_{i2} \neq \ldots \neq C_{im}$ thì f1, f2, ..., fn độc lập logic

ĐỘC LẬP LOGIC (tt)

VÍ DỤ 2: Xét tính độc lập logic của f1, f2

$$\begin{cases} f1: & AB \vee \overline{B} \\ f2: & A\overline{B} \end{cases}$$

Với

A:	0	1	0	1
B:	0	0	1	1
AB:	0	0	0	1
$\overline{B}$	1	1	0	0
$A\overline{B}$	0	1	0	0

#f1: 1 1 0 1

#f2: 0 1 0 0

Giá trị các cột:(1 3 0 1)➔ #F (1 1 0 1) = #f1

NHẬN XÉT: Thứ tự sắp xếp các hàng (fi) rất quan trọng ➔ **độc lập logic không có tính giao hoán.**

Ví dụ: (#f1 #f2 #f3) ≠ (#f2 #f3 #f1)

PHƯƠNG TRÌNH LOGIC

*** QUI ƯỚC:**

- Mệnh đề ẩn : X, Y, ...
- Mệnh đề hằng : A, B,C, ...

*** VÍ DỤ1:** Giải phương trình logic:

X (A + B) = ABC .... (*)

Có nghiệm: $x = f(A, B, C) = \begin{cases} 1/\ ABC \\ 2/\ ABC + \bar{A}\bar{B}\bar{C} \\ 3/\ ABC + \bar{A}\bar{B}C \\ 4/\ \bar{A}\bar{B} + AB \end{cases}$

Giải phương trình (*) đưa về dạng:

#X (A + B) = #ABC

#X (#(A + B)) = #ABC

#X (#A + #B) = #A #B #C ... (**)

PHƯƠNG TRÌNH LOGIC (tt)

Có 3 mệnh đề → cần $2^3 = 8$ bit #A: 0101 0101

#B: 0011 0011 #C: 0000 1111

(**) ⇔ #X (0 1 1 1 0 1 1 1) = 0000 0001

Vậy # X có dạng: ? ! 0 0 0 ! 0 0 1 với ! = 0 hoặc 1

$$\Rightarrow \begin{cases} 0000\ 0001 \\ 0000\ 1001 \\ 1000\ 0001 \\ 1000\ 1001 \end{cases} \Rightarrow \begin{cases} X1 = ABC \\ X2 = ABC + \overline{A}\,\overline{B}\,\overline{C} \\ X3 = ABC + \overline{A}\,\overline{B}C \\ X4 = \overline{A}\,\overline{B} + AB \end{cases}$$

Dùng tuyển chuẩn hay hội chuẩn suy ra nghiệm

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + ABC$$

PHƯƠNG TRÌNH LOGIC (tt)

VÍ DỤ 2: Bài tóan đĩa bay UFO

Cho A: xuất hiện vệt sáng

B: xuất hiện mây

C: xuất hiện vệt khói

X: xuất hiện đĩa bay

Y: xuất hiện vật khác

- Phỏng vấn 1: $\overline{AXY} + \overline{BX} + C + AB$
- Phỏng vấn 2: $A + BX\overline{Y} + CY$
- Suy ra:

$$A + BX\overline{Y} + CY = \overline{AXY} + \overline{BX} + C + AB$$

$$AI + BX\overline{Y} + CY = \overline{AXY} + \overline{BX} + C + ABI$$

PHƯƠNG TRÌNH LOGIC (tt)

- Tính $\#F_{VT}$
- Tìm X = f1(A, B, C) và Y = f2(A, B, C)

Suy ra tìm #X và #Y trong không gian ABC

#X: $\underbrace{0\ 1\ .....\ 1}_{8}$ và #Y: $\underbrace{1\ 0\ .....\ 1}_{8}$

- **# F trái = # F phải** tại những vị trí giống nhau của 2 ma trận

	1	2	3	4	5	6	7	8
#X	1,1	0,0						
#Y	0,1	0,1						
	2	2	2	4	2	4	3	4

→ **có 3072 khả năng**

- Nếu lấy phần tử đầu tiên ta được 1 nghiệm

#X = 1000 0010 và #Y = 0010 1000

- **Để được 1 nghiệm bất lỳ cứ lấy 1 cột ghép với những cột khác (một cột → một vị trí)**

PHƯƠNG TRÌNH LOGIC (tt)

VÍ DỤ:

$$\begin{cases} (A + B)\,X + AY = A & \dots (1) \\ Y + X = A + B & \dots (2) \end{cases}$$

Nghiệm $C_0C_1C_2C_1$

$C_0C_1C_2C_3$

$C_0C_3C_2C_1$

$C_0C_3C_2C_3$

ĐỀ TÀI:

❶ **Tìm các điều kiện để suy ra f → g theo #f và #g**

Chứng minh rằng: #f ⊕ #g = #f

Thì f → g ⇔ f ∨ g

➁ **Tính độc lập logic: các điều kiện để F(f1 f2 ... fn) = I**

➂ **Lấy ví dụ một bài toán hệ phương trình logic** và giải hệ phương trình logic

LOGIC XÁC SUẤT
– LOGIC KHẢ SUẤT

LOGIC XÁC SUẤT (tt)

*** BÀI TÓAN:**

$$\left.\begin{array}{l} P(A) \\ P(\rightarrow) \end{array}\right\} \rightarrow P(B)\ ? \qquad \left.\begin{array}{l} P(A) \\ P(B) \end{array}\right\} \rightarrow P(\rightarrow)\ ?$$

*** HẠN CHẾ:**

P(A) quá bé thì không xảy ra

Biết P(A), P(B) nhưng không suy ra được P(A + B)

*** HỆ THỨC CỦA LOGIC XÁC SUẤT**

$$\left.\begin{array}{l} Pr(p \rightarrow q) \geq a \\ Pr(q) \geq b \end{array}\right\} \Rightarrow Pr(q) \geq \max(0, a + b - 1)$$

LOGIC KHẢ SUẤT

$Pr(A) + Pr(\overline{A}) = 1 \rightarrow Pos(A)?$

$Pos(A) > Pr(A)$

$Pos(A) < Pr(A)$

Cr(q): credit của q (độ tin cậy / độ xác tín của q)

$$Cr(q) = \sum_{\forall p \rightarrow q} m(p)$$ mà $$\sum_{\forall p} m(p) = 1$$

Suy ra

$$Cr(q) + Cr(\neg q) \leq 1$$

- **ĐỘ ĐO CÓ THỂ TIN (Plausible)**

$Pl(p) = 1 - Cr()$

Ví dụ: Cho A → P(A) = 0.7

Cr(A) = 0.6 ; Pl(A) = 0.8

LỚP HÀM KHẢ SUẤT ĐẶC BIỆT

* **ĐỘ ĐO NEC (NECCESSITY)**
  - NEC(A) = Cr(A)
  - NEC(A $\cap$ B) = min (NEC(A) , NEC(B))
* **ĐỘ ĐO POS**
  - Pos A = Pl(A)
  - Pos (A $\cap$ B) = max (Pos A, Pos B)

LỚP HÀM KHẢ SUẤT ĐẶC BIỆT (tt)

TÍNH CHẤT

❶ **Min (Nec(p) , Nec()) = 1**

Min (Nec(p) , Nec($\overline{p}$)) = Nec(p $\wedge \overline{p}$) (theo định nghĩa)

= Nec (false)

= 0 (đúng)

❷ ...

❸ **Pos(P) < 1 → Nec(P) > 0**

❹ **Nec(P) > 0 → Pos(P) =1**

Vì theo ❶ → Nec($\overline{p}$) = 0

Pos(p) = 1 – Nec($\overline{p}$) = 1 (đúng)

❺ ...

LỚP HÀM KHẢ SUẤT ĐẶC BIỆT (tt)

TÍNH CHẤT

1. $\left.\begin{array}{ll} Pr(\rightarrow) & \geq a \\ Pr(p) & \geq b \end{array}\right\}$ $\quad Pr(q) \geq \max(0, a+b-1)$

2. $\left.\begin{array}{ll} Nec(p\rightarrow q) & \geq a \\ Nec(p) & \geq b \end{array}\right\}$ $\quad Nec(q) \geq \min(a,b)$

3. $\left.\begin{array}{ll} Nec(p\rightarrow q) & \geq a \\ Nec(q) & \leq b \end{array}\right\}$ $\quad Nec(p) \leq \begin{cases} 1, & \text{nếu } a \leq b \\ b, & \text{nếu } a > b \end{cases}$

4. $\left.\begin{array}{ll} Nec(p\rightarrow q) & \geq a \\ Pos(p) & \geq b \end{array}\right\}$ $\quad Pos(q) \geq \begin{cases} 0, & \text{nếu } a + b \leq 1 \\ b, & \text{nếu } a+b>1 \end{cases}$

LỚP HÀM KHẢ SUẤT ĐẶC BIỆT (tt)

TÍNH CHẤT

- $\left.\begin{array}{l} Nec(p \rightarrow q) \geq a \\ Pos(q) \leq b \end{array}\right\} \quad Pos(p) \leq \max(1-a,b)$

- $\left.\begin{array}{l} Pos(p \rightarrow q) \geq a \\ Nec(p) \geq b \end{array}\right\} \quad Pos(q) \leq \begin{cases} 0, & \text{nếu } a+b \leq 1 \\ a, & \text{nếu } a+b > 1 \end{cases}$

LOGIC KHẢ SUẤT NHIỀU CHIỀU

- **Khái niệm**
  - Đánh giá cấp độ tin cậy của các suy diễn dựa trên luật và khái luật
  - Biểu diễn độ tin cậy của luật p→q bằng ma trận khả suất
- Gọi π là độ đo khả suất
- Ta có ma trận cột đánh giá p là:

$$\begin{pmatrix}\Pi(p)\\ \Pi(\neg p)\end{pmatrix} \begin{array}{l}\longleftarrow \text{Khả năng xảy ra}\\ \longleftarrow \text{Khả năng không xảy ra}\end{array}$$

- *Ví dụ*

$$\pi(p) = \begin{pmatrix}0.8\\ 0.3\end{pmatrix}$$

LOGIC KHẢ SUẤT NHIỀU CHIỀU

- Khả năng xảy ra và khả năng không xảy ra của một sự kiện không nhất thiết phải bằng 1
- Trường hợp đặc biệt
  - Nếu p là sự kiện gần như xảy ra (chắc chắn) có thể viết theo cách sau,

$$\Pi(p) = \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}$$ với ε dùng để chỉ mức độ xảy ra

  - Nếu p là sự kiện chắc chắn xảy ra tuyệt đối; có thể viết theo cách sau,

$$\Pi(p) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

LOGIC KHẢ SUẤT NHIỀU CHIỀU (tt)

- Ma trận cột đánh giá luật p→q là:

$$\begin{pmatrix} \Pi(p \to q) & \Pi(\bar{p} \to q) \\ \Pi(p \to \bar{q}) & \Pi(\bar{p} \to \bar{q}) \end{pmatrix} \text{ hay } \Pi(p \to q) = \begin{pmatrix} 1 & \varepsilon_1 \\ \varepsilon_2 & \varepsilon_3 \end{pmatrix}$$

Với ε1 < ε2 ; ε1 < ε3 ; ε2 < ε3; ε1,ε2, ε3 < 0.01

$$\Pi(z) = \begin{pmatrix} 1 \\ \tau \end{pmatrix} \quad \text{với} \quad \tau = \frac{\varepsilon_1 + \varepsilon_2}{\varepsilon_3}$$

Để giải quyết bài toán cho ứng dụng này cần biết công thức cơ bản sau:

$$\Pi(Q) = \Pi(P \to Q) * \Pi(P)$$

$$\Leftrightarrow \begin{pmatrix} \Pi(Q) \\ \Pi(\bar{Q}) \end{pmatrix} = \begin{pmatrix} \Pi(P \to Q) & \Pi(\bar{P} \to Q) \\ \Pi(P \to \bar{Q}) & \Pi(\bar{P} \to \bar{Q}) \end{pmatrix} * \begin{pmatrix} \Pi(P) \\ \Pi(\bar{P}) \end{pmatrix}$$

LOGIC KHẢ SUẤT NHIỀU CHIỀU (tt)

- **Qui ước:**

$$A*B = \min(A,B)$$

$$A + B = \max(A,B)$$

- **Bài toán:** Giả sử có một số sự kiện

1. *Nếu Bảo đến thì nói chung Mai đến*
2. *Bảo đến*
3. *Mai đến thì Thắng đến*
4. *Nếu Sơn đến thì Mai không đến*
5. *Thường là Sơn đến*

Logic hình thức chỉ được dựa trên những luật chắc chắn.

Áp dụng độ đo xác suất một chiều để xem độ chắc chắn của từng phát biểu như thế nào

LOGIC KHẢ SUẤT NHIỀU CHIỀU (tt)

GIẢI: Giả sử $\lambda<\tau$

1. $B \rightsquigarrow M$ $\qquad \prod(B \rightsquigarrow M) = \begin{pmatrix} 1 & ? \\ \lambda & ? \end{pmatrix}$
2. B $\qquad \prod(B) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
3. $M \rightarrow T$ $\qquad \prod(M \rightarrow T) = \begin{pmatrix} 1 & ? \\ 0 & ? \end{pmatrix}$
4. $S \rightarrow \overline{M}$ $\qquad \prod(S \rightarrow \overline{M}) = \begin{pmatrix} ? & 1 \\ ? & 0 \end{pmatrix}$
5. $\rightsquigarrow S$ $\qquad \prod(\rightsquigarrow S) = \begin{pmatrix} 1 \\ \tau \end{pmatrix}$

LOGIC KHẢ SUẤT NHIỀU CHIỀU (tt)

Đánh giá khả năng Mai đến:

$$\begin{pmatrix} \Pi(M) \\ \Pi(\neg M) \end{pmatrix} = \Pi(B \to M) * \Pi(B)$$

$$= \begin{pmatrix} 1 & ? \\ \lambda & ? \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ \lambda \end{pmatrix}$$

LOGIC KHẢ SUẤT NHIỀU CHIỀU (tt)

Đánh giá khả năng Thắng đến dựa vào 3 luật 1,2,3:

$$\begin{pmatrix}\Pi(T)\\ \Pi(\neg T)\end{pmatrix} = \Pi(M \to T) * \Pi(M)$$

$$= \begin{pmatrix}1 & ?\\ 0 & ?\end{pmatrix} * \begin{pmatrix}1\\ \lambda\end{pmatrix} = \begin{pmatrix}1*1+?*\lambda\\ 0*1+?*\lambda\end{pmatrix}$$

$$= \begin{pmatrix}1\\ [0,\lambda]\end{pmatrix}$$

(luật Thắng đến)

$$\begin{pmatrix}\Pi(T)\\ \Pi(\neg T)\end{pmatrix} = \begin{pmatrix}\Pi(M \to T) & \Pi(\neg M \to T)\\ \Pi(M \to \neg T) & \Pi(\neg M \to \neg T)\end{pmatrix} * \begin{pmatrix}\Pi(S \to M) & \Pi(\neg S \to M)\\ \Pi(S \to \neg M) & \Pi(\neg S \to \neg M)\end{pmatrix}$$

$$= \begin{pmatrix}1 & ?\\ 0 & ?\end{pmatrix} * \begin{pmatrix}\mu & ?\\ 1 & ?\end{pmatrix} * \begin{pmatrix}1\\ \delta\end{pmatrix}$$

$$= \begin{pmatrix}[\mu,1]\\ [0,1]\end{pmatrix}$$

SỐ MỜ - LOGIC MỜ
- GIẢI PHƯƠNG TRÌNH MỜ

1. Tập mờ

- Định nghĩa: A là tập mờ trên không gian nền X nếu A được xác định bởi hàm:

$$\mu A: X \rightarrow [0,1]$$

Trong đó:

μA là hàm liên thuộc (membership function)

$\mu A(x)$ là độ liên thuộc của x vào tập mờ A.

2. Các phép toán trên tập mờ

- Định nghĩa: Cho A và B là hai tập mờ trên không gian nền X, có các hàm liên thuộc μA, μB. Khi đó ta có các phép toán sau:

Stt	Phép toán trân tập mờ	Định nghĩa hàm liên thuộc
1	$A \subseteq B$	$\mu A(x) \leq \mu B(x)$
2	$A \cup B$	$\mu A \cup B(x) = \max\{\mu A(x), \mu B(x)\}$
3	$A \cap B$	$\mu A \cap B(x) = \min\{\mu A(x), \mu B(x)\}$
4	$\neg A$	$\mu \neg A = 1 - \mu A$
5	$A \oplus B$	$\mu A \oplus B = \mu A(x) + \mu B(x) - \mu A(x) \mu B(x)$
6	X	$\mu X(x) = 1$
7	ϕ	$\mu \phi(x) = 0$
8	AxB	$\mu AxB(x,y) = \min\{\mu A(x), \mu B(y)\}$

Số mờ

Định nghĩa: Tập mờ M trên đường thẳng số thực R^l là tập số mờ nếu:

a) M là chuẩn hoá, tức là có điểm x’ sao cho $\mu M(x')=1$.

b) ứng với mỗi $\alpha \in R^l$, tập mức $\{x:\mu M(x)\geq\alpha\}$là đoạn đóng trên R^l

Người ta thường dùng các số mờ dạng tam giác, hình thang và dạng Gauss

SỐ MỜ (tt)

- **CỘNG SỐ MỜ:**

  [a, b] + [d, e] = [a + d, b + e]

- **TRỪ SỐ MỜ:**

  [a, b] - [d, e] = [a - e , b - d]

- **NHÂN SỐ MỜ:**

  [a, b] * [d, e] = [min (ad, ae, bd, be]), max (ad, ae, bd, be)]

- **CHIA SỐ MỜ:**

[a, b] / [d, e] = [min (a/d, a/e, b/d, b/e]), max (a/d, a/e, b/d, b/e)]

Nguyên lý suy rộng của Zadeh

- Để làm việc với các hệ thống có nhiều biến vào, nguyên lý suy rộng của Zadeh là rất quan trọng.

- Định nghĩa: Cho A_i là tập mờ với các hàm liên thuộc μAi trên không gian nền Xi, (i=1,2,…,n). Khi ấy tích của A1xA2x…xAn là tập mờ trên không gian nền X=X1xX2x…xXn với hàm liên thuộc:

$$\mu A(x)=\min\{\mu A_1(x_1),\mu A_2(x_2),\ldots,\mu A_n(x_n)\}$$

Trong đó $x=(x_1,x_2,\ldots,x_n)$

Nguyên lý suy rộng

- Giả sử với mỗi biến đầu vào xi lấy giá trị là Ai(i=1,2,..,n) với Ai là tập mờ trên không gian nền Xi và hàm liên thuộc là μAi(xi). Hàm f: X→Y chuyển các giá trị đầu vào Ai thánh giá trị đầu ra B. Khi đó B sẽ là tập mờ trên Y với các hàm liên thuộc μB(x) được tính theo công thức sau:

$\mu B(x) = \max\{\min(\mu A_1(x_1),\ldots,\mu A_n(x_n)) : x \in f^{-1}(y)\}$ nếu $f^{-1}(y) \neq \phi$

$\mu B(x) = 0$ nếu $f^{-1}(y) = \phi$

Trong đó $f^{-1}(y) = \{x = (x_1, x_2, \ldots, x_n) \in X : f(x) = y\}$

Suy rộng phép cộng hai số mờ

Áp dụng nguyên lý suy rộng chúng ta có thể cho ngay định nghĩa suy rộng phép cộng cho 2 số mờ bằng cách sử dụng hàm hai biến. z= f(x,y)= x+y

Định nghĩa: Cho M,N là 2 số mờ có hàm liên thuộc μM(x), μN(x) khi đó cộng suy rộng M+N là tập mờ trên R' có hàm liên thuộc xác địnhvới mỗi số thực z cho bởi: μM(x) +N(z)={min(μM(x) ,μN(y)) : x+y= z}

Định lý: (Dubois, Prade 1980**)** Nếu M,N là 2 số mờ hình thang thì M+N cũng là số mờ hình thang.

Tương tự người ta cũng định nghĩa phéptrừ suy rộng và phép nhân suy rộng

SỐ HỌC MỜ

- Số học mờ dựa trên hai tính chất của con số mờ:
  - Mỗi tập mờ cũng như mỗi số mờ có thể được nêu ra bởi α-cuts đầy đủ và duy nhất.
  - α-cuts của số mờ là khoảng đóng thực $\forall\alpha \in (0, 1]$
- **Gọi * là một trong 4 {+, -, ., /}**

  $[a,b]*[d,e]=\{f \times g \,/\, a \le f \le b,\ d \le g \le e\}$

- Những hoạt động số học mờ dựa trên khoảng đóng.

Gọi $A=[a_1, a_2]$, $B=[b_1, b_2]$, $C=[c_1, c_2]$, $O=[o_1, o_2]$, $1=[1,1]$ ta có:

SỐ HỌC MỜ

1. **A+B = B+A, A.B=B.A**
2. **(A+B)+C = A+(B+C), (A.B).C=A.(B.C)**
3. **A = 0+A = A+0, A=1.A=A.1**
4. **A.(B+C) $\subseteq$ A.B+A.C**
5. **Nếu b.c $\geq$ 0 $\forall b \in B, \forall c \in C$ thì A.(B.C)=A.B+A.C**
6. **0$\in$A-A và 1$\in$A/A**
7. **Nếu A $\subseteq$ E và B$\subset$ F thì:**
   - **A+B $\subseteq$ E+F**
   - **A-B $\subseteq$ E-F**
   - **A.B $\subseteq$ E.F**
   - **A/B $\subseteq$ E/F**

Logic mờ

Để giảm bớt sự phụ thuộc vào cácphép tính min max, do đó làm tăng mềm dẻo và linh hoạt tron việc giải các bài toán thực tế người ta mở rộng phép lấy min max thành 2 lớp t-norm và t-conorm có từng cặp phần tử đối ngẫu.

Như vậy chúng ta sẽ không có một đại số tập mờ duy nhất, vì trong định nghĩa đại số tập mờ ta luôn có thể thay min , max bằng t-norm và t-conorm đối ngẫu nhau và thu được một tập đại số mờ khác

1. Phép phủ định

Định nghĩa: Hàm $n:[0,1]\rightarrow[0,1]$ không tăng thỏa mãn điều kiện $n(0)=1$, $n(1)=0$, gọi là hàm phủ định.

Một vài ví dụ:

- Hàm phủ định chuẩn $n(x)=1-x$
- Hàm phủ định $n(x)=1-x^2$
- Họ phủ định (Sugeno,1997) $N_\lambda(x)=(1-x)/(1+\lambda x)$, với $\lambda>-1$

Hàm n là phép phủ định mạnh, nếu n giảm chặt và $n(n(x))=x$ mọi x.

2. Phép hội (t-norm)

Phép hội (vẫn quen gọi là phép AND) là 1 trong các phép toán logic cơ bản nhất. Nó cũng là cơ sở để định nghĩa phép giao của 2 tập mờ.

<u>Định nghĩa:</u> Hàm $T:[0,1]^2 \rightarrow [0,1]$ là một t-norm nếu thỏa mãn các điều kiện sau:

a) $T(1,x)=x$, với mọi $0 \leq x \leq 1$ (Tồn tại phần tử đơn vị)

b) $T(x,y)=T(y,x)$, với mọi $0 \leq x, y \leq 1$ (T có tính giao hoán)

c) $T(x,y)=T(u,v)$, với mọi $0 \leq x \leq u \leq 1$, $0 \leq y \leq v \leq 1$

(Không giảm theo từng biến)

d) $T(x,T(y,z))=T(T(x,y),z)$ với mọi $0 \leq x,y,z \leq 1$ (T có tính kết hợp)

Từ những tiêu đề trên chúng ta suy ra ngay $T(0,x)$. Hơn nữa tiên đề d) đảm bảo tính thác triển duy nhất cho hàm nhiều biến.

Một vài ví dụ :

- Min (Zadeh 1965) : $T(x,y)=\min(x,y)$
- Dạng tích : $T(x,y)=xy$
- t-norm Lukasiewiz : $T(x,y)=\max\{x+y-1,0\}$
- Min nilpotent (Fodor 1993): $T(x,y)=\min(x,y)$ nếu $x+y>1$

  $=0$ nếu $x+y\leq 1$
- t-norm yếu nhất (drastic, product):

  $Z(x,y)=\min(x,y)$ nếu $\max(x,y)=1$

  $=0$ nếu $\max(x,y)<1$

Không khó khăn để chỉ ra rằng với mỗi 1-norm T thì

$Z(x,y)\leq T(x,y)\leq \min(x,y)$ với mọi $0\leq x,y\leq 1$

3. Phép tuyển (t-conorm)

Giống như phép hội, phép tuyển (hay tóan tử OR) thông thường cần thỏa mãn các tiên đề sau:

Định nghĩa: Hàm S:[0,1] gọi là phép tuyển (OR suy rộng) hay là t-conorm nếu thỏa mãn các tiên đề sau:

a) $S(0,x)=x$, với mọi $0\leq x\leq 1$ (Tồn tại phần tử đơn vị)

b) $S(x,y)=S(x,y)$, với mọi $0\leq x,y\leq 1$ (S có tính giao hoán)

c) $S(x,y)\leq S(u,v)$, với mọi $0\leq x\leq u\leq 1$, $0\leq y\leq v\leq 1$
(Không giảm theo từng biến)

d) $S(x,S(y,z))=S(S(x,y),z)$ với mọi $0\leq x,y,z\leq 1$ (S có tính kết hợp)

Định lý: Cho n là 1 phép phủ định mạnh, T là một t-norm, khi ấy hàm S xác định trên: [0,1]2 bằng biểu thức S(x,y)=nT(nx,ny) với mọi 0≤x,y≤1, là một t-conorm. Chọn phép phủ định n(x)=1-x chúng ta có quan hệ giữa T và S như trong bảng:

T(x,y)	S(x,y)
min(x,y)	max(x,y)
xy	x+y-xy
max(x+y-1, 0)	min(x+y, 1)
min_0(x,y)=min(x,y) nếu x+y>1 =0 nếu max(x,y)≤1	max1(x,y)=max(x,y) nếu x+y<1 =0 nếu min(x,y)>0
Z(x,y)=min(x,y) nếu max(x,y)=1 =0 nếu max(x,y)<1	Z'(x,y)=max(x,y) nếu min(x,y)=0 =0 nếu min(x,y)>0

4. Luật De Morgan

Định nghĩa: cho T là t-norm, S là t-conorm, và n là phép phủ định chặt. Chúng ta nói bộ ba (T,S,n) là một bộ ba De Morgan nếu n(S(x,y))=T(nx,ny)

Chúng ta nói bộ ba (T,S,n) là liên tục nếu T và S là hai hàm liên tục. Sau đây là hai lớp bộ ba quan trọng

Định nghĩa: bộ ba De Morgan (T,S,n) là bộ ba mạnh (strong) khi và chỉ khi có một tự đồng cấu φ: $[0,1]\to[0,1]$ sao cho:

a) $T(x,y)= \varphi^{-1}(\max(\varphi(x) + \varphi(y)-1, 0))$

b) $S(x,y) = \varphi^{-1}(\min(\varphi(x) + \varphi(y), 1))$

c) $N(x) = \varphi^{-1}(1- \varphi(x))$

Định nghĩa: bộ ba De Morgan (T,S,n) là bộ ba chặt (strict) khi và chỉ khi có một tự đồng cấu φ: $[0,1]\to[0,1]$ sao cho:

a) $T(x,y)= \varphi^{-1}((\varphi(x)\varphi(y))$

b) $S(x,y) = \varphi^{-1}((\varphi(x) + \varphi(y) - \varphi(x)\varphi(y))$

c) $N(x) = \varphi^{-1}(1- \varphi(x))$

5. Phép kéo theo

Đã có khá nhiều nghiên cứu về phép kéo theo. Điều đó cũng tự nhiên vì đây là công đoạn mấu chốt của quá trình suy diễn trong mọi lập luận xấp xỉ, bao gồm cả suy luận mờ.

Định nghĩa: Phép kéo theo (implication) là một hàm số I: $[0,1] \rightarrow [0,1]$ thỏa mãn các điều kiện sau:

a) Nếu $x \leq z$ thì $I(x,y) \geq I(z,y)$ với mọi $y \in [0,1]$

b) Nếu $y \leq u$ thì $I(x,y) \leq I(x,u)$ với mọi $y \in [0,1]$

c) $I(0,x) = 1$ với mọi $x \in [0,1]$

d) $I(x,1) = 1$ với mọi $x \in [0,1]$

e) $I(1,0) = 0$

Tuy đơn giản nhưng mục e) vẫn cần vì không thể suy ra mục e) từ 4 tiên đề đầu.

Để tính toán được, chúng ta cần những dạng cụ thể của phép kéo theo. Sau đây là một số dạng hàm kéo theo, xây dựng dựa vào các phép toán logic mờ đã suy rộng trên

Cho T là t-norm, S là t-conorm, và n là hàm phủ định mạnh

Định nghĩa: Dạng kéo theo thứ nhất. Hàm $I_{s1}(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức $I_{s1}(x,y)= S(n(x),y)$

Rõ ràng ẩn ý sau định nghĩa này là công thức logic cổ điển $P \Rightarrow Q = \rceil P \vee Q$.

Định lý: Với bất kỳ t-norm T, t-conorm S và phép phủ định mạnh n nào, I_{S1} là phép kéo theo thỏa định nghĩa phép kéo theo.

Phép kéo theo thứ hai sau đây lấy từ lôgic trực cảm (intuitionistic logic)

Định nghĩa: Cho T là t-norm, hàm $I_T(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức

$$I_T(x,y)= \sup\{u:T(x,u)\leq y\}$$

Định lý: Với bất kỳ t-norm T nào, I_T được định nghĩa như trên là phép kéo theo thỏa định nghĩa phép kéo theo.

PHƯƠNG TRÌNH MỜ

Xét 2 dạng phương trình mờ:

- **Dạng 1: A + X = B**

Không thể suy ra X = B – A,

Thật vậy, với $A=[a_1,a_2]$, $B=[b_1,b_2]$, $B-A=[b_1-a_1,b_2-a_2]$

$A+B = [a_1,a_2] + [b_1-a_2,b_2-a_1] = [a_1+b_1-a_2, a_2+b_2-a_1] \neq [b_1,b_2] = B$

Đặt $X=[x_1,x_2]$, ta có $[a_1+x_1, a_2+x_2] = [b_1,b_2]$

$$\begin{cases} a_1 + x_1 = b_1 \\ a_2 + x_2 = b_2 \end{cases} \qquad \begin{cases} x_1 = b_1 - a_1 \\ x_2 = b_2 - a_2 \end{cases}$$

Vì X là 1 khoảng với $x_1 \leq x_2$ → $b_1 - a_1 \leq b_2 - a_2$ thì có nghiệm

Lúc đó nghiệm là **$x=[b_1-a_1,b_2-a_2]$**

PHƯƠNG TRÌNH MỜ (tt)

* ***<u>Cách giải quyết 2:</u>***

$\forall \alpha \in (0,1]$, $\alpha_A = [\alpha_{a1}, \alpha_{a2}]$, $\alpha_B = [\alpha_{b1}, \alpha_{b2}]$ và $\alpha_X = [\alpha_{x1}, \alpha_{x2}]$

phương trình có nghiệm $\Leftrightarrow$

i/ $\alpha_{b1} - \alpha_{a1} \leq \alpha_{b1} - \alpha_{a2}, \forall \alpha \in (0,1]$

ii/ $\alpha \leq \beta \Rightarrow \alpha_{b1} - \alpha_{a1} \leq \beta_{b1} - \beta_{a1} \leq \beta_{b2} - \beta_{a2} \leq \alpha_{b2} - \alpha_{a2}$

$$\Rightarrow x = \bigcup_{\alpha \in (0,1]} \alpha X$$

PHƯƠNG TRÌNH MỜ (tt)

- **Dạng 2: A. X = B**

Đặt $\alpha_A=[\alpha_{a1},\alpha_{a2}]$, $\alpha_B=[\alpha_{b1},\alpha_{b2}]$ và $\alpha_X=[\alpha_{x1},\alpha_{x2}]$ thì nghiệm phương trình mờ tồn tại, tương đương với

$$\begin{cases} \dfrac{\alpha_{b_1}}{\alpha_{a_1}} \le \dfrac{\alpha_{b_2}}{\alpha_{a_2}}, \forall \alpha \in [0,1] \\ \alpha \le \beta \rightarrow \dfrac{\alpha_{b_1}}{\alpha_{a_1}} \le \dfrac{\beta_{b_1}}{\beta_{a_1}} \le \dfrac{\beta_{b_2}}{\beta_{a_2}} \le \dfrac{\alpha_{b_2}}{\alpha_{a_2}} \end{cases}$$

Nếu nghiệm phương trình tồn tại

$$\Longrightarrow X = \bigcup_{\alpha \in [0,1]} \alpha X$$

CÁC PHÉP TÓAN TRÊN TẬP MỜ

Gọi A, B là hai tập mờ trong tập X

- **Phép hợp:**

$$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x))$$

- **Phép giao:**

$$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x))$$

- **Phép bù:**

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

- **Phép kéo theo:**

$$\mu_{A\rightarrow B}(x) = \mu_{\bar{A}\cup B}(x) = \max(1 - \mu_A(x), \mu_B(x))$$

HỆ LUẬT MỜ

- **HỆ LUẬT MỜ**

  IF ... THEN

- Theo luật rõ: **If t > 30, a > 15 then v = 300**
- **Nguyên lý xử lý các bài tóan mờ: Input rõ thì sẽ mờ hoá để áp dụng luật**

  rõ → mờ

- Dữ liệu vào rõ → mờ hóa để tìm luật áp dụng → từ đó rõ hóa để áp dụng. Rõ → mờ → rõ

HỆ LUẬT MỜ

- Nhật là quốc qua đầu tiên ứng dụng hệ luật mờ để điều khiển tàu điện ngầm ở Tokyo → ra đời thế hệ máy móc thông minh – intelligent machine → system

HT + IT

(high tech + information technology)

- *Ví dụ:* ứng dụng trong hoạt động của các máy giặt
  - Nếu quần áo bẩn + nhiều thì xà phòng nhiều + máy quay lâu
  - Nếu quần áo sạch+ nhiều thì xà phòng ít+ máy quay lâu
  - Nếu quần áo bẩn + ít thì xà phòng nhiều + máy quay vừa
  - ……

→ có 36 luật

MỘT VÀI ỨNG DỤNG THỰC TẾ

Ngày nay lý thuyết tập mờ được ứng dụng rộng rãi trong lĩnh vực điều khiển tự động và hệ chuyên gia.

Chúng ta khảo sát một vài bài toán nhỏ có ứng dụng thực tế từ các khái niệm về tập mờ.

Bài Toán 1: Bài Toán Lọc Khách Hàng Trùng

1. Mô tả:

Một công ty bán hàng khi bán hàng có nhập các thông tin của khách vào cơ sở dữ liệu. Do lượng khách hàng lớn và giảm thời gian tìm kiếm dữ liệu nên mỗi lần khách mua hàng công ty lại nhập lại thông tin trên → dữ liệu bị trùng. Cuối năm công ty muốn tặng lịch và gửi thiệp chúc tết tới tất cả khách hàng và không muốn 1 khách hàng nhận cùng lúc nhiều tấm thiệp (giống nhau). Do đó công ty phải gom tất cả khách hàng trùng lại. Tuy nhiên do có nhiều người nhập khác nhau dễ có sai sót trong qua trình nhập (thêm bớt khoảng trắng, bỏ dấu và không bỏ dấu, nhập sai chình tả, nhập thiếu thông tin, …).

Bài toán đặt ra là cần có một công cụ có thể gom các mẩu tin nghi ngờ trùng của khách hàng để dễ xử lý khách hàng trùng. Việc gom dữ liệu trùng này là rất cần thiết, rất có lợi trong việc phân tích dữ liệu kinh doanh

2. Phân tích bài toán:

1. Các dữ liệu liên quan:

Trong dữ liệu khách hàng để phân biệt trùng ta cần thu thập các thông tin về Tên, địa chỉ, mã số thuế (nếu có). Trong phần địa chỉ lại được tách ra thành tên đường, số nhà, phường/xã, quận/huyện. vậy ta cần quan tâm tất cả là 6 trường trong cơ sở dữ liệu.

Ta sẽ coi 1 record trong cơ sở dữ liệu làm chuẩn và tìm các record khác nghi ngờ giống nó, lập lại công việc trên cho toàn bộ các record trong cơ sở dữ liệu ta có được kết quả mong muốn.

2. Vấn đề cần giải quyết

Bài toán đặt ra là ta cần phải đưa ra các định nghĩa về độ giống nhau của các thông tin khách hàng. Điều này hoàn toàn giống với khái niệm về liên thuộc trong tập mờ.

Các trường dữ liệu đều là chuỗi nên các phép toán dựa trên so sánh chuỗi.

Việc xây dựng hàm liên thuộc cho từng trường dữ liệu rồi chọn 1 hàm t-norm để tính hàm liên thuộc chung là giao của các hàm liên thuộc gặp một khó khăn là các trường có mức ưu liên khác nhau trong việc đánh giá độ giống nhau của hai record dữ liệu khách hàng => quyết định dùng công thức tính tổng trung bình có trọng số để ra hàm liên thuộc.

3. Thiết kế và xây dựng ứng dụng

Xây dựng hàm đánh giá độ giống nhau trong dữ liệu (vd : trường tên)

Bước 1: Thực hiện chuỗi cần so sánh bằng cách bỏ các khoảng trắng thừa (vẫn dữ lại 1), upcase lại chuỗi.

Bước 2: So sánh 2 chuỗi và tìm số vị trí khác nhau N.\

Bước 3: Tính hàm lượng giá f = max(0 , 1-0.1*N). Trong Trường hợp có 1 chuỗi không được nhập (giá trị NULL), ta quy định f = 0.5

Bước 4: Tính hàm liên thuộc $\mu(x) = \Sigma\ \alpha i * fi$

Trong đó fi là các giá trị tính được trong bước 3 và các số αi là các hệ số tương ứng với các trường dữ liệu. Ta có $\Sigma\ \alpha i = 1$

Bảng hệ số đề nghị

Tên trường	Hệ số
Họ tên	0.35
Mã số thuế	0.35
Số nhà	0.05
Tên đường	0.15
Phường / xã	0.05
Quận / huyện	0.05

Thực Hiện Truy vấn: thực hiện câu truy vấn dữ liệu có sử dụng hàm của câu 1 với một lượng giá (vd : f > 0.5) ta sẽ truy xuất được tập thông tin khách hàng nghi ngờ trùng => vấn đề của bài toán.

4. Kết quả và các vấn đề còn lại

- Kết quả là trả về tập các khách hàng nghi ngờ trùng, việc chọn câu lệnh với hàm f khá lớn (>0.9) sẽ cho những khách hàng có khả năng trùng rất cao.
- Khi so sánh các chuỗi ký tự ta chưa đề cập tới vấn đề đồng âm (những từ có cách phát âm gần giống dễ gây nhầm lẫn) và việc nhập có bỏ dấu hay không bỏ dấu, nếu cải tiến được các điều này kết quả sẽ chính xác hơn.
- Tùy thuộc vào tính chất của cơ sở dữ liệu ta có thể chọn được các hệ số α tốt cũng như lượng giá tốt trường hợp nhập bị Null sẽ cho ta một hàm đánh giá tốt hơn.
- Bài toàn trên có được từ ứng dụng khái niệm của hàm liên thuộc, bài toán tiếp theo dựa trên khái niệm ứng dụng số mờ và phép toán trên tập mờ.

Bài toán 2: Bài toán phân tích dữ liệu marketing

1. Mô tả:

Thông thường 1 công ty muôn thiết lập kế hoạch sản xuất phải ước lượng được nhu cầu tiêu dùng của khách hàng cũng như sở thích của khách hàng. Họ thường phát các phiếu thăm dò khách hàng để thu thập thông tin. Thông tin thăm dò sẽ bao gồm thông tin khách hàng (thông tin cá nhân, thông tin nghề nghiệp, mức sống trình độ …) thông tin về sản phẩm (đặc tình sản phẩm, đối thủ cạnh tranh, nhu cầu đối với sản phẩm của khách hàng, …) cho khách hàng chọn lựa và đánh giá. Dựa trên việc phân tích dữ liệu thu được mà đưa ra kế hoạch tối ưu.

Bài toán đặt ra là thông tin của khách hàng không phải luôn chính xác do đó cần phải có 1 công cụ có thể lượng giá được độ chính xác của thông tin khách hàng, lý thuyết về tập mờ và đại số gia tử khá phù hợp để giải quyết bài toán này.

2. Phân ch

-

n.

-

c nhau.

-

c.

-

n.

-

p.

3. Thiết kế và xây dựng ứng dụng:

- Biểu diễn thông tin lựa chọn của khách hàng theo số mờ dạng tam giác.
- Thực hiện lấy giá trị tổng hợp bằng lấy lượng trung bình (trung bình cận trái, giá trị trung bình, trung bình cận phải).
- Thực hiện xây dựng các hệ số để từ đó tạo ra được mối liên quan giữa dữ liệu phân tích với kết quả thực tế.
- Khi có nhiều dữ liệu liên quan nhau ta có thể dùng một phép kéo theo mở rộng để suy diễn theo logic mờ.

4. Kết quả và các vấn đề còn lại:

- Bài toán lấy mẫu số liệu và dự đoán gặp phải trở ngại là trong thực tế có những thành phần ảnh hưởng mà ta chưa lường hết được, tuy nhiên nếu có thể thực hiện được nhiều lần theo phương pháp thử sai thì hệ thống sẽ dần đạt đến trạng thái có thể đưa ra một kết qua tốt.

- Bài toán này cũng hơi giống với Case Study 1 trong sách tham khảo.