5.7 Explanation-Based Learning (EBL)

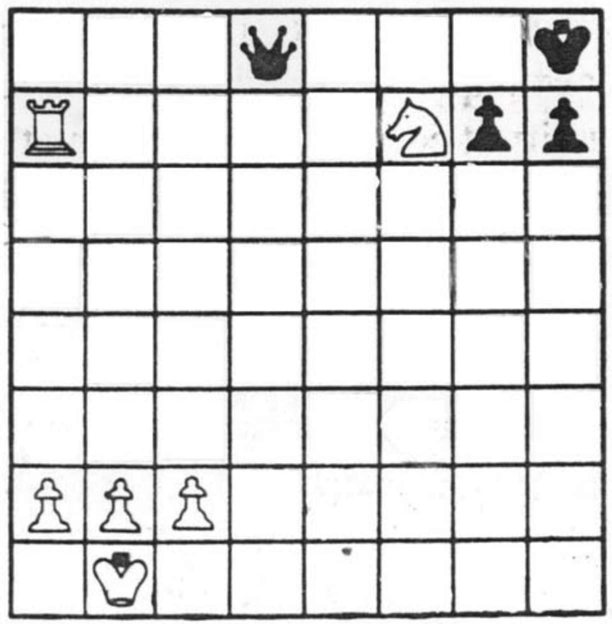
- เป็นวิธีเรียนรู้ซึ่งเรียนจากตัวอย่างบวกเพียงตัวเดียว
- เช่น การเรียน concept "fork" ในการเล่น chess

  white knight attacks both the black king and black queen

  ในกรณีนี้ฝ่ายดำต้องยอมเสีย queen

  จากตัวอย่างเดียว สิ่งที่เรียนรู้คือ

  if any piece x attacks both the opponent's king and another piece y, then y will be lost.
- ใช้ domain-specific knowledge ช่วยในการเรียน
- กระบวนการของ EBL
  - ใช้ domain knowledge อธิบายว่าทำไมตัวอย่างจึงเป็นตัวอย่างของ concept ในรูปของกฎ
  - generalize กฎที่ได้ เพื่อให้ใช้กับกรณีอื่นได้



A Fork Position in Chess

Input & output ของ EBL

Input:

- Training example -- ตัวอย่างของ concept ที่จะเรียน (board position ที่แสดง fork)
- Goal concept -- concept ที่จะเรียน (concept fork)
- Operational criterion -- description ที่สามารถนำไปใช้ได้ทันที

  (attack-both(WKn,BK,BQ) ไม่สามารถนำไปใช้ได้ทันที ต้องแสดงในรูปของตำแหน่งบนกระดาน เช่น

  position(WKn,f7), position(BK,h8), position(BQ,d8))
- Domain theory -- กฎต่าง ๆที่ใช้แสดงความสัมพันธ์ของ objects และ actions ใน domain นั้น (กฎการเล่น chess)

Output

- generalization ของ training example ซึ่งเพียงพอสำหรับอธิบาย goal concept และสอดคล้องกับ operational criterion

การเรียน concept "cup"

- Training example:

  owner(object23,ralph), has-part(object23,concavity12),
  isa(concavity12,concavity), is(concavity12,upward-pointing),
  has-part(object23,handle16), isa(handle16,handle),
  is(object23,light), color(object23,brown), has-part(object23,bottom19),
  is(bottom19,bottom), is(bottom19,flat), . . .
- Domain theory

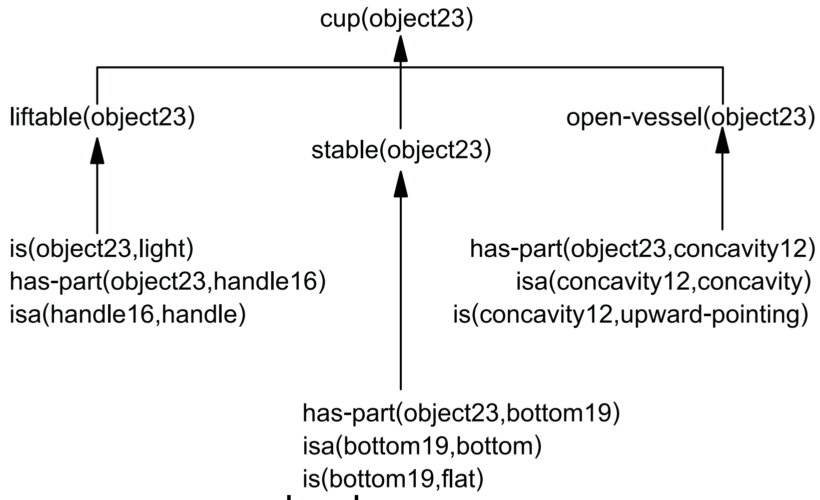
  liftable(X), stable(X), open-vessel(X) → cup(X)
  is(X,light), has-part(X,Y), isa(Y,handle) → liftable(X)
  small(X), made-from(X,Y), low-density(Y) → liftable(X)
  has-part(X,Y), isa(Y,bottom), is(Y,flat) → stable(X)
  has-part(X,Y), isa(Y,concavity), is(Y,upward-pointing)
  → open-vessel(X)
- Goal concept : *cup(X)*

  X is a cup iff X is liftable, stable and open-vessel.
- Operational criterion : *สิ่งที่แสดงลักษณะต่าง ๆของ cup เช่น isa, has-part, color*

- จากตัวอย่างบวกที่ให้มา เราต้องการสร้าง description ของ cup

(1) ใช้ domain knowledge อธิบายว่าทำไม object23 จึงเป็น cup

สร้าง proof tree ของ object23



สังเกตุว่า predicate ที่ไม่เกี่ยวข้องกับ concept เช่น owner, color จะไม่มีใน proof tree

(2) generalize และ ดึง predicate ที่เป็น operational criterion มาสร้างกฎ

- generalization ตาม domain knowledge

  ถ้า argument ของ predicate ที่ตรงกันใน domain knowledge เป็น variable ก็เปลี่ยน argument ที่เป็น constant ให้เป็น variable

  ถ้า argument ของ predicate ที่ตรงกันใน domain knowledge เป็น constant ก็ไม่ต้องเปลี่ยน

เช่น is(object23,light) เปลี่ยนเป็น is(X,light)

โดยที่ X แทน object23

- ดึง predicate ที่เป็น operational criterion มาสร้างกฎได้

  is(X,light),has-part(X,H), isa(H,handle), has-part(X,B), isa(B,bottom), is(B,flat), has-part(X,C), isa(C,concavity), is(C,upward-pointing) → cup(X)

5.10 Bayesian Learning

- Bayesian learning เป็นเทคนิคที่ใช้ทฤษฎีความน่าจะเป็น (Bayes theorem) เพื่อหาว่าสมมติฐาน (hypothesis)ใด น่าจะถูกต้องที่สุด
- ใช้ความรู้ก่อนหน้า (prior knowledge) ร่วมกับข้อมูล (data) เพื่อหาสมมติฐานที่ดีที่สุด
  - ความน่าจะเป็นก่อนหน้าสำหรับสมมติฐานหนึ่งๆ
  - ความน่าจะเป็นของข้อมูลที่สังเกตุได้สำหรับสมมติฐานหนึ่งๆ
- เป็นเทคนิคที่ใช้งานจริงได้ดี เช่น naive Bayes learning, Bayesian belief network learning

Bayes Theorem

- $P(h \mid D) = \dfrac{P(D \mid h) * P(h)}{P(D)}$

  โดยที่ $P(h \mid D)$ คือ ความน่าจะเป็นของ h เมื่อรู้ D

  $P(D \mid h)$ คือ ความน่าจะเป็นของ D เมื่อรู้ h

  $P(h)$ คือ ความน่าจะเป็นก่อนหน้าของสมมติฐาน h

  $P(D)$ คือ ความน่าจะเป็นก่อนหน้าของเซตของตัวอย่าง D

- ใช้ Bayes theorem ในการหาสมมติฐานที่น่าจะเป็นที่สุดเมื่อรู้ D

  Maximum a posteriori hypothesis : h_{MAP}

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D) = \arg\max_{h \in H} \frac{P(D \mid h) * P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D \mid h) * P(h)$$

- Maximum Likelihood (h_{ML}) ($P(h_i)=P(h_j)$) $= \arg\max_{h \in H} P(D \mid h)$

ตัวอย่างการเลือกสมมติฐานโดย Bayes Theorem

- คนไข้คนหนึ่งไปตรวจหามะเร็ง ผลการตรวจเป็นบวก
  - ผลการตรวจเมื่อเป็นบวกจะให้ความถูกต้อง 98% ของกรณีที่มีโรคนั้นอยู่จริง
  - ผลการตรวจเมื่อเป็นลบจะให้ความถูกต้อง 97% ของกรณีที่ไม่มีโรคนั้น
  - นอกจากนั้น 0.008 ของประชากรทั้งหมดเป็นโรคมะเร็ง
- คนไข้คนนี้เป็นมะเร็งจริงหรือไม่?

$P(cancer) =$ $\qquad$ $P(\neg cancer) =$

$P(+ \mid cancer) =$ $\qquad$ $P(- \mid cancer) =$

$P(+ \mid \neg cancer) =$ $\qquad$ $P(- \mid \neg cancer) =$

$h_{MAP} =$

สูตรพื้นฐานเกี่ยวกับความน่าจะเป็น

- Product Rule : ความน่าจะเป็น $P(A \wedge B)$ ที่สองเหตุการณ์ A และ B จะเกิดพร้อมกัน

  $P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$
- Sum Rule : ความน่าจะเป็น $P(A \vee B)$ ที่เหตุการณ์ A หรือ B จะเกิด

  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
- Theorem of total probability : ถ้าเหตุการณ์ $A_1,\ldots,A_n$ ไม่เกิดร่วมกัน และ $\sum_{i=1}^{n} P(A_i) = 1$ แล้ว

  $P(B) = \sum_{i=1}^{n} P(B \mid A_i)P(A_i)$
- Chain Rule : $A_1, A_2, \ldots, A_n$ เป็นเหตุการณ์ n เหตุการณ์

  $P(A_1, A_2, \ldots, A_n) = \sum_{i=1}^{n} P(A_i \mid A_{i-1}, \ldots, A_1)$

การแยกแยะที่น่าจะเป็นที่สุดสำหรับตัวอย่าง

- h_{MAP} เป็นสมมติฐานที่น่าจะเป็นที่สุด แต่ไม่เป็นการแยกแยกตัวอย่างที่น่าจะเป็นที่สุด (most probable classification)
- พิจารณาสมมติฐานทั้งสามต่อไปนี้

  $P(h_1 | D) = 0.4 \quad P(h_2 | D) = 0.3 \quad P(h_3 | D) = 0.3$
- เมื่อให้ตัวอย่าง x ผลการแยกแยะของสมมติฐานเป็นดังนี้

  $h_1(x) = + \qquad h_2(x) = - \qquad h_3(x) = -$
- การแยกแยะตัวอย่าง x ที่น่าจะเป็นที่สุดคือ?

Bayes Optimal Classifier

- ตัวแยกแยะที่ดีที่สุดแบบเบย์ :

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i)\, P(h_i \mid D)$$

- ตัวอย่าง

$P(h_1 | D) = 0.4 \qquad P(- \mid h_1) = 0 \qquad P(+ \mid h_1) = 1$

$P(h_2 | D) = 0.3 \qquad P(- \mid h_2) = 1 \qquad P(+ \mid h_2) = 0$

$P(h_3 | D) = 0.3 \qquad P(- \mid h_3) = 1 \qquad P(+ \mid h_3) = 0$

$$\Rightarrow \quad \sum_{h_i \in H} P(+ \mid h_i)\, P(h_i \mid D) = 0.4$$

$$\sum_{h_i \in H} P(- \mid h_i)\, P(h_i \mid D) = 0.6$$

- ดังนั้น $\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i)\, P(h_i \mid D) = -$

Naive Bayes Classifier

- ตัวแยกแยะเบย์อย่างง่าย (naive Bayes classifier) เป็นตัวแยก-แยะประเภทหนึ่งที่ใช้งานได้ดี เหมาะกับกรณีของ
  - เซตตัวอย่างมีจำนวนมาก
  - คุณสมบัติ (attribute) ของตัวอย่างไม่ขึ้นต่อกัน
- ประยุกต์ใช้งานในด้าน
  - การแยกแยะข้อความ (text classification)
  - การวินิจฉัย (diagnosis)

Naive Bayes Classifier

- ให้ $a_1, a_2, \ldots, a_n$ เป็นคุณสมบัติของตัวอย่าง
- ค่าที่น่าจะเป็นที่สุดของตัวอย่าง x คือ

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, \ldots, a_n)$$

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n \mid v_j) * P(v_j)}{P(a_1, a_2, \ldots, a_n)}$$

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n \mid v_j) * P(v_j)$$

- Naive Bayes assumption :

$$P(a_1, a_2, \ldots, a_n \mid v_j) = \prod_i P(a_i \mid v_j)$$

- Naive Bayes classifier : $v_{NB} = \arg\max_{v_j \in V} P(v_j) * \prod_i P(a_i \mid v_j)$

Naive Bayes Algorithm

- Naive_Bayes_Learn(examples)

  For each target value v_j

  $\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

  For each attribute value a_i of each attribute a

  $\hat{P}(a_i \mid v_j) \leftarrow$ estimate $P(a_i \mid v_j)$

- Classify_New_Example(x)

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) * \prod_i \hat{P}(a_i \mid v_j)$$

Name	Hair	Height	Weight	Lotion	Result
Sarah	blonde	average	light	no	+
Dana	blonde	tall	average	yes	–
Alex	brown	short	average	yes	–
Annie	blonde	short	average	no	+
Emily	red	average	heavy	no	+
Pete	brown	tall	heavy	no	–
John	brown	average	heavy	no	–
Katie	blonde	short	light	yes	–

- สมมติว่าตัวอย่างที่ต้องการแยกแยะคือ

  Judy blonde average heavy no class = ?

- คำนวณ $v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) * \prod_i \hat{P}(a_i \mid v_j)$

  P(+)P(blonde|+)P(average|+)P(heavy|+)P(no|+) = 1/18

  P(–)P(blonde|–)P(average|–)P(heavy|–)P(no|–) = 1/125

  $\Rightarrow v_{NB} = +$

Learning To Classify Text

Target concept *Interest?* : Document $\rightarrow$ {+, –}

1. Represent each document by vector of words
   - one attribute per word position in document
2. Learning: Use training examples to estimate
   - P(+)
   - P(–)
   - P(doc | +)
   - P(doc | –)

Naive Bayes conditional independence assumption

$$P(doc \mid v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k \mid v_j)$$

- $P(a_i = w_k \mid v_j)$: ความน่าจะเป็นที่คำในตำแหน่งที่ i เป็น w_k เมื่อรู้ v_j
- สมมติฐานเพิ่มเติม $P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j), \forall i,m$

Learn_naive_Bayes_text

Learn_naive_Bayes_text(*Examples,V*)

1. Collect all words and other tokens that occur in *Examples*

- *Vocabulary* $\leftarrow$ all distinct words and other tokens in *Examples*

2. Calculate the required $P(v_j)$ and $P(w_k \mid v_j)$

- For each target value v_j in *V* do
  - -- $docs_j \leftarrow$ subset of *Examples* for which the target value is v_j
  - -- $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - -- $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - -- $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)

Learn_naive_Bayes_text

-- for each word w_k in *Vocabulary*

$n_k \leftarrow$ number of times word w_k occurs in $Text_j$

$$P(w_k \mid v_j) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$

Classify_naive_Bayes_text(*Doc*)

- *positions* $\leftarrow$ all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{NB} where

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) * \prod_{i \in positions} P(a_i \mid v_j)$$

Bayesian Belief Networks

- Naive Bayes assumption มีข้อจำกัดมากเกินไป
- Bayesian belief network ใช้อธิบายความไม่ขึ้นต่อกันอย่างมีเงื่อนไข (conditional independent) ระหว่างตัวแปร
- ทำให้เราสามารถใช้

(1) ความรู้ก่อนหน้า (prior knowledge) เกี่ยวกับ ความ(ไม่)ขึ้นต่อกันระหว่างตัวแปร,

ร่วมกับ

(2) ตัวอย่าง

- บางครั้งเราเรียกว่า Bayes Net

Conditional Independence

- Definition: X is conditionally independent of Y given Z if the probability distribution governing X is independent of the value of Y given the value of Z; that is, if

  $(\forall\ x_i, y_j, z_k)\ P(X=x_i \mid Y=y_j, Z=z_k) = P(X=x_i \mid Z=z_k)$

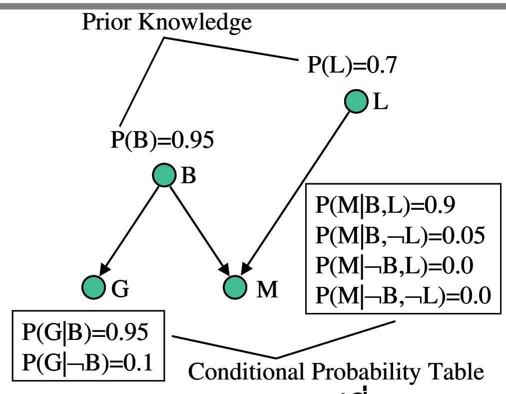
  or simply,

  $P(X \mid Y,Z) = P(X \mid Z)$

- Example: Thunder is conditionally independent of Rain, given Lightning

  P(Thunder | Rain, Lightning) = P(Thunder | Lightning)

Bayesian Belief Network



B : Battery is charged

L : block is Lifetable

M : arm Moves

G : Gauge indicates that battery is charged

รูปที่ 5.10.1

- Bayes net แสดงเซตของความ (ไม่) ขึ้นต่อกันของตัวแปร
- แสดงโดย Directed Acyclic Graph (DAG)
- แต่ละโนดไม่ขึ้นต่อกันอย่างมีเงื่อนไข (conditional independent) กับโนดอื่น เมื่อรู้โนดพ่อแม่โดยตรง (immediate predecessors)

Bayesian Belief Network

- แสดงการกระจายความน่าจะเป็นร่วม (joint probability distribution) ระหว่างตัวแปร
  - เช่น P(Battery,Liftable,Gauge,Move)
- โดยทั่วไป

$$P(y_1,\ldots,y_n) = \prod_{i=1}^{n} P(y_i \mid \mathrm{Parents}(Y_i))$$

โดยที่ $\mathrm{Parents}(Y_i)$ หมายถึงโนดพ่อแม่โดยตรงของโนด Y_i

- การกระจายความน่าจะเป็นร่วมนิยามโดยกราฟและ $P(y_i \mid \mathrm{Parents}(Y_i))$
- จากตัวอย่าง $P(G,M,B,L) = P(G|B,M,L)P(M|B,L)P(B|L)P(L)$

  $= P(G|B)P(M|B,L)P(B)P(L)$

รูปแบบการอนุมานใน Bayes Net (1)

1. Causal Reasoning : การอนุมานจากเหตุ

   เช่น ต้องการคำนวณ P(M|L) -- หาความน่าจะเป็นที่แขนจะเคลื่อนได้เมื่อรู้ว่ากล่องยกได้ (กล่องยกได้เป็นสาเหตุหนึ่งของการที่แขนจะเคลื่อนได้)

   - กระจาย P(M|L) ให้อยู่ในรูปของผลรวมของความน่าจะ–เป็นร่วมระหว่าง M กับโนดพ่อแม่อื่นนอกจาก L (M กับ B)

     $P(M|L) = P(M,B|L) + P(M,\neg B|L)$

   - จัดรูปให้ M ขึ้นกับโนดพ่อแม่ (B,L) โดยใช้ chain rule

     $P(M|L) = P(M|B,L)P(B|L) + P(M|\neg B,L)P(\neg B|L)$

     $\Rightarrow P(M|L) = P(M|B,L)P(B) + P(M|\neg B,L)P(\neg B)$

     $= 0.855$

รูปแบบการอนุมานใน Bayes Net (2)

2. Diagnosis Reasoning : การอนุมานจากผล

เช่น ต้องการคำนวณ $P(\neg L|\neg M)$ -- ความน่าจะเป็นที่กล่องยกไม่ได้เมื่อรู้ว่าแขนไม่ได้เคลื่อน (ใช้ผล -- อาการ เพื่อหาสาเหตุ)

$$P(\neg L|\neg M) = \frac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)} \qquad \text{(Bayes' rule)}$$

- คำนวณ $P(\neg M|\neg L)$ ได้ 0.9525 (ใช้ causal reasoning)
- คำนวณ $P(\neg L|\neg M) = \frac{0.9525 * 0.3}{P(\neg M)} = \frac{0.28575}{P(\neg M)}$
- ในทำนองเดียวกัน $P(L|\neg M) = \frac{P(\neg M|L)P(L)}{P(\neg M)}$

$$= \frac{0.145*0.7}{P(\neg M)} = \frac{0.1015}{P(\neg M)}$$

- $P(\neg L|\neg M) + P(L|\neg M) = 1 \Rightarrow P(\neg L|\neg M) = 0.7379$

รูปแบบการอนุมานใน Bayes Net (3)

3. Explaining away : ทำ causal reasoning ภายใน diagnostic reasoning

- เช่น เมื่อรู้ $\neg M$ (แขนไม่เคลื่อน) เราสามารถคำนวณ $\neg L$ (ความน่าจะเป็นที่กล่องไม่สามารถยกได้)
- แต่ถ้าเรารู้ $\neg B$ (แบตเตอรีไม่ได้ชาร์จ) แล้ว $\neg L$ ควรจะมีค่าความน่าจะเป็นน้อยลง
- ในกรณีนี้เราเรียกว่า $\neg B$ explain $\neg M$, making $\neg L$ less certain

$$P(\neg L|\neg B,\neg M) = \frac{P(\neg M,\neg B|\neg L)P(\neg L)}{P(\neg B,\neg M)} \qquad \text{(Bayes' rule)}$$

$$= \frac{P(\neg M|\neg B,\neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B,\neg M)}$$

$$= \frac{P(\neg M|\neg B,\neg L)P(\neg B)P(\neg L)}{P(\neg B,\neg M)}$$

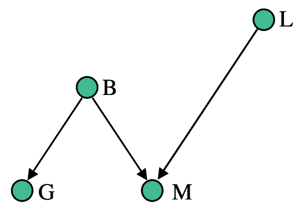
- หลังคำนวณ $P(\neg B,\neg M)$ เราได้ $P(\neg L|\neg B,\neg M) = 0.30$

การเรียนรู้ Bayes Net

- การเรียนรู้ Bayes Net คือการหาเน็ตเวิร์ก (structure และ conditional probability tables) ที่สอดคล้องกับตัวอย่างมากที่สุด
- ปัญหาการเรียนรู้แบ่งเป็น
  - (1) structure unknown
  - (2) structure known
    - (2.1) no missing value data
    - (2.2) with missing value data
- กรณีของ structure known + no missing value data เป็นกรณีที่ง่ายที่สุด สามารถทำการเรียนรู้ได้ในลักษณะเดียวกับ naive Bayes classifer

Structure Known with No Missing Value Data (Learn CPTs)

- คำนวณ CPT สำหรับแต่ละโนด
- ต้องใช้ตัวอย่างจำนวนมากเพื่อให้ค่าทางสถิติถูกต้อง



รูปที่ 5.10.2

G	M	B	L	no. of instances
True	True	True	True	54
True	True	True	False	1
True	False	True	True	7
True	False	True	False	27
False	True	True	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2
				100

- $\hat{P}(V_i = v_i \mid \text{Parents}(V_i) = \mathbf{p}_i) = \dfrac{\text{จำนวนตัวอย่างที่มี } V_i = v_i}{\text{จำนวนตัวอย่างที่มี Parents}(V_i) = \mathbf{p}_i}$
- $P(B=true) = (54+1+7+27+3+2)/100 = 0.94$
- $P(L=true) = (54+7+3+4)/100 = 0.68$
- $P(M|B,\neg L)$ = อัตราส่วนที่ M=true เมื่อ B=true, L=false $= 1/(1+27+2) = 0.03$
- เราสามารถคำนวณ CPT สำหรับโนด G ได้ในทำนองเดียวกัน

Structure Known with Missing Value Data (Learn CPTs)

- ตัวอย่างเช่นข้อมูลต่อไปนี้ โดยที่ * แทน missing value

G	M	B	L	no. of instances
True	True	True	True	54
True	True	True	False	1
*	*	True	True	7
True	False	True	False	27
False	True	*	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2

รูปที่ 5.10.3

- พิจารณากรณีของ 3 ตัวอย่างที่มีค่า G=false, M=true, L=true ในกรณีนี้เราไม่รู้ค่าของ B แต่อาจคำนวณ P(B| ¬ G,M,L) หรือ P(¬ B|¬ G,M,L) ได้ ถ้าเรารู้ CPT (แต่เรายังไม่รู้)
- จากนั้น เราจะแทนที่ตัวอย่างทั้งสามด้วยตัวอย่างมีน้ำหนัก (weighted examples) 2 ตัว
  - ตัวแรกคือตัวอย่างที่ B=True มีน้ำหนัก= P(B| ¬ G,M,L)
  - ตัวที่สองคือตัวอย่างที่ B=False มีน้ำหนัก= P(¬ B|¬ G,M,L)

Structure Known with Missing Value Data (Learn CPTs)

- ในทำนองเดียวกัน กรณีของ 7 ตัวอย่างที่มีค่า B=true, L=true และ G และ M ไม่รู้ค่านั้น เราสามารถแทนที่ตัวอย่างทั้งเจ็ดด้วยตัวอย่างมีน้ำหนัก (weighted examples) 4 ตัว
  - (1) ตัวอย่าง G=true, M=true มีน้ำหนัก = P(G,M|B,L)
  - (2) ตัวอย่าง G=true, M=false มีน้ำหนัก = P(G,¬ M|B,L)
  - (3) ตัวอย่าง G=false, M=true มีน้ำหนัก = P(¬ G,M|B,L)
  - (4) ตัวอย่าง G=false, M=false มีน้ำหนัก = P(¬ G,¬ M|B,L)
- ในทำนองเดียวกัน เราสามารถค่าความน่าจะเป็นเหล่านี้ได้ ถ้าเรารู้ CPTs และ structure ของ Bayes Net
- จากนั้นเราจะใช้ตัวอย่างมีน้ำหนักเหล่านี้ ร่วมกับตัวอย่าง-ที่เหลือเพื่อคำนวณ CPTs ได้ (ตัวอย่างที่ไม่รู้ค่าถูกแทนที่ด้วยตัวอย่างมีน้ำหนัก)

Expectation-Maximization (EM) Algorithm

- อัลกอริทึม EM สามารถใช้เพื่อคำนวณค่าน้ำหนักของตัวอย่างที่ไม่รู้ค่าได้ (คำนวณ CPTs)
  (1) เริ่มต้นโดย เราสุ่มค่าให้กับพารามิเตอร์ทั้งหมดของ CPTs
  (2) เราใช้ CPTs เพื่อคำนวณน้ำหนัก (ความน่าจะเป็นของค่าต่างๆของตัวอย่างที่ไม่รู้ค่า เมื่อรู้ตัวอย่างอื่นๆ)
  (3) เราใช้น้ำหนักที่คำนวณได้ เพื่อประมาณ CPTs ใหม่
  (4) ทำซ้ำ (2) - (3) จนกระทั่ง CPTs ลู่เข้า
- โดยทั่วไปอัลกอริทึม EM จะใช้เวลาในการลู่เข้าไม่มาก

ตัวอย่างของอัลกอริทึม EM (1)

พิจารณาตัวอย่างในรูปที่ 5.10.3

(1) สุ่มค่าสำหรับตาราง CPTs

- P(L) = 0.5 (P(¬ L) = 1– P(L))
- P(B) = 0.5 (P(¬ B) = 1– P(B))
- P(M|B,L) = 0.5 (P(¬ M|B,L) = 1 – P(M|B,L))
  P(M|B,¬ L) = 0.5 (P(¬ M|B,¬ L) = 1 – P(M|B,¬ L))
  P(M|¬ B,L) = 0.5 (P(¬ M|¬ B,L) = 1 – P(M|¬ B,L))
  P(M|¬ B,¬ L) = 0.5 (P(¬ M|¬ B,¬ L) = 1 – P(M|¬ B,¬ L))
- P(G|B) = 0.5 (P(¬ G|B) = 1 – P(G|B))
  P(G|¬ B) = 0.5 (P(¬ G|¬ B) = 1 – P(G|¬ B))

ตัวอย่างของอัลกอริทึม EM (2)

(2) ใช้ CPTs เพื่อคำนวณน้ำหนัก

- ตัวอย่างที่ไม่รู้ค่าคือ

G	M	B	L	no. of instances
*	*	True	True	7
False	True	*	True	3

- ในกรณีของ 7 ตัวอย่างแรกเราต้องการหา $P(G,M|B,L)$, $P(G,\neg M|B,L)$, $P(\neg G,M|B,L)$ และ $P(\neg G,\neg M|B,L)$
  - $P(G,M|B,L) = P(G|B)*P(M|B,L) = 0.5*0.5$
  - $P(G,\neg M|B,L) = P(G|B)*P(\neg M|B,L) = 0.5*0.5$
  - $P(\neg G,M|B,L) = P(\neg G|B)*P(M|B,L) = 0.5*0.5$
  - $P(\neg G,\neg M|B,L) = P(\neg G|B)*P(\neg M|B,L) = 0.5*0.5$

ตัวอย่างของอัลกอริทึม EM (3)

- แสดงว่า 7 ตัวอย่างแรก เราสามารถใส่น้ำหนักให้เป็นตัวอย่างดังต่อไปนี้

G	M	B	L	no. of instances
True	True	True	True	7*0.5*0.5=1.75
True	False	True	True	7*0.5*0.5=1.75
False	True	True	True	7*0.5*0.5=1.75
False	False	True	True	7*0.5*0.5=1.75

- ในกรณีของ 3 ตัวอย่าง

G	M	B	L	no. of instances
False	True	*	True	3

เราต้องการหา $P(B|\neg G,M,L)$ และ $P(\neg B|\neg G,M,L)$

- $$P(B|\neg G,M,L) = \frac{P(B,\neg G,M,L)}{P(\neg G,M,L)}$$

$$= \frac{P(\neg G|B,M,L)*P(M|B,L)*P(B|L)*P(L)}{P(\neg G,M,L,B) + P(\neg G,M,L,\neg B)}$$

ตัวอย่างของอัลกอริทึม EM (4)

$$= \frac{P(\neg G|B,M,L)*P(M|B,L)*P(B|L)*P(L)}{P(\neg G|B,M,L)*P(M|B,L)*P(B|L)*P(L) + P(\neg G|\neg B,M,L)*P(M|\neg B,L)*P(\neg B|L)*P(L)}$$

$$= \frac{P(\neg G|B)*P(M|B,L)*P(B)}{P(\neg G|B)*P(M|B,L)*P(B) + P(\neg G|\neg B)*P(M|\neg B,L)*P(\neg B)}$$

- ดังนั้น $P(B|\neg G,M,L) = (0.5*0.5*0.5)/(0.5*0.5*0.5+0.5*0.5*0.5) = 0.5$
  $P(\neg B|\neg G,M,L) = 0.5$
- แสดงว่า 3 ตัวอย่าง เราสามารถใส่น้ำหนักให้เป็นตัวอย่างดังต่อไปนี้

G	M	B	L	no. of instances
False	True	True	True	3*0.5=1.5
False	True	False	True	3*0.5=1.5

ตัวอย่างของอัลกอริทึม EM (5)

- ตัวอย่างทั้งหมด

G	M	B	L	no. of instances
True	True	True	True	54
True	True	True	False	1
True	*True*	*True*	*True*	*1.75*
True	*False*	*True*	*True*	*1.75*
False	*True*	*True*	*True*	*1.75*
False	*False*	*True*	*True*	*1.75*
True	False	True	False	27
False	*True*	*True*	*True*	*1.5*
False	*True*	*False*	*True*	*1.5*
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2

ตัวอย่างของอัลกอริทึม EM (6)

(3) ใช้ตัวอย่างมีน้ำหนักที่คำนวณได้ เพื่อประมาณ CPTs ใหม่

- $P(L) = 68/100 = 0.680$ $(P(\neg L) = 1 - P(L))$
- $P(B) = 92.5/100 = 0.925$ $(P(\neg B) = 1 - P(B))$
- $P(M|B,L) = 59/62.5 = 0.944$ $(P(\neg M|B,L) = 1 - P(M|B,L))$
  $P(M|B,\neg L) = 1/30 = 0.033$ $(P(\neg M|B,\neg L) = 1 - P(M|B,\neg L))$
  $P(M|\neg B,L) = 1.5/5.5 = 0.273$ $(P(\neg M|\neg B,L) = 1 - P(M|\neg B,L))$
  $P(M|\neg B,\neg L) = 0/2 = 0.000$ $(P(\neg M|\neg B,\neg L) = 1 - P(M|\neg B,\neg L))$
- $P(G|B) = 85.5/92.5 = 0.924$ $(P(\neg G|B) = 1 - P(G|B))$
  $P(G|\neg B) = 0/7.5 = 0.000$ $(P(\neg G|\neg B) = 1 - P(G|\neg B))$

ตัวอย่างของอัลกอริทึม EM (7)

(2) ใช้ CPTs เพื่อคำนวณน้ำหนักของตัวอย่างไม่รู้ค่าใหม่

- ในกรณีของ 7 ตัวอย่างแรก
  - $P(G,M|B,L) = P(G|B)*P(M|B,L) = 0.924*0.944 = 0.872$
  - $P(G,\neg M|B,L) = P(G|B)*P(\neg M|B,L) = 0.924*0.056 = 0.052$
  - $P(\neg G,M|B,L) = P(\neg G|B)*P(M|B,L) = 0.076*0.944 = 0.072$
  - $P(\neg G,\neg M|B,L) = P(\neg G|B)*P(\neg M|B,L) = 0.076*0.056 =0.004$
- ได้ตัวอย่างมีน้ำหนักเป็น

G	M	B	L	no. of instances
True	True	True	True	7*0.872=6.11
True	False	True	True	7*0.052=0.36
False	True	True	True	7*0.072=0.50
False	False	True	True	7*0.004=0.03

ตัวอย่างของอัลกอริทึม EM (8)

- ในกรณีของ 3 ตัวอย่าง
  - $P(B|\neg G,M,L) = \dfrac{0.076*0.944*0.925}{0.076*0.944*0.925 + 1.000*0.273*0.075} = 0.764$
  - $P(\neg B|\neg G,M,L) = 1- 0.764 = 0.236$
- แสดงว่า 3 ตัวอย่าง เราสามารถใส่น้ำหนักให้เป็นตัวอย่างดังต่อไปนี้

G	M	B	L	no. of instances
False	True	True	True	3*0.764=2.29
False	True	False	True	3*0.236=0.71

- เมื่อทำซ้ำขั้นตอน (2), (3) จนครบ 20 รอบ CPTs ลู่เข้าดังนี้ (<0.001)

$P(L) = 0.680$ $P(B) = 0.940$

$P(M|B,L) = 1.000$ $P(M|B,\neg L) = 0.033$

$P(M|\neg B,L) = 0.005$ $P(M|\neg B,\neg L) = 0.000$

$P(G|B) = 0.943$ $P(G|\neg B) = 0.000$