ปัญญาประดิษฐ์ (Artificial Intelligence)

บทบรรยายที่ 6: การให้เหตุผลเกี่ยวกับเวลา
(Reasoning over Tiการแจกแจงร่วม ของตัวแปรทั้งหมดจนถึงเวลา t คือ:

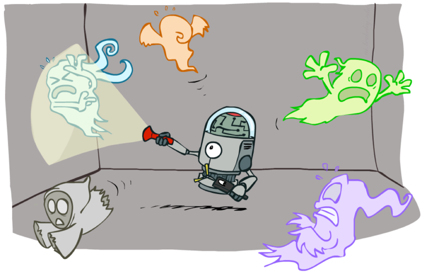
$$\mathbf{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i | \mathbf{X}_{i-1}) \mathbf{P}(\mathbf{E}_i | \mathbf{X}_i)$$

ผศ. ดร. อิทธิพล ฟองแก้ว
[ittipon@g.sut.ac.th]

เนื้อหาวันนี้

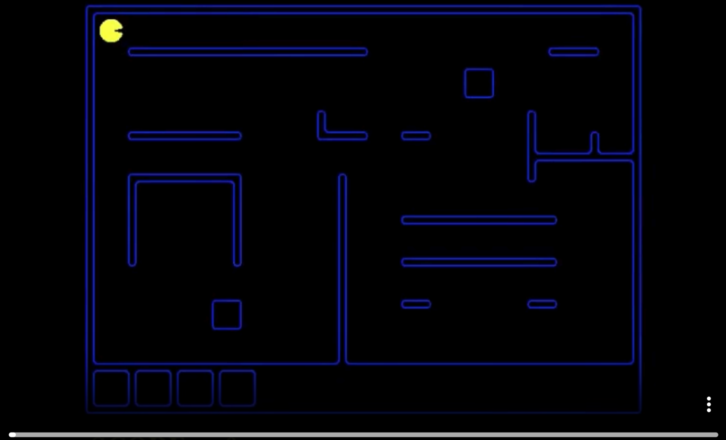
การรักษา สถานะความเชื่อ (belief state) เกี่ยวกับโลก และปรับปรุงข้อมูลเมื่อเวลาผ่านไปและมีหลักฐานใหม่เข้ามา

- Markov models
  - Markov processes
  - งานการ inference
  - Hidden Markov models
- Filters
  - Kalman filter
  - Particle filter



บทบรรยายนี้สำคัญมาก อย่ามองข้าม!

—
เครดิต: CS188, UC Berkeley.



Pacman ผู้แก้แค้น: จะใช้ข้อมูลจาก sonar readings ให้เกิดประโยชน์ได้อย่างไร?

Markov Models

แบบจำลองที่พิจารณาความเปลี่ยนแปลงตามเวลา

การจำลองการเปลี่ยนแปลงของเวลา

เราจะพิจารณาโลกเป็น ชุดเวลาแบบไม่ต่อเนื่อง (discrete time slices) โดยแต่ละช่วงเวลาจะประกอบด้วยตัวแปรสุ่ม:

- $\mathbf{X}_t$ หมายถึงชุดของ ตัวแปรสถานะที่ไม่สามารถสังเกตได้ ณ เวลา t
- $\mathbf{E}_t$ หมายถึงชุดของ ตัวแปรหลักฐานที่สังเกตได้ ณ เวลา t

แผนผังแสดงความสัมพันธ์ตามเวลา

```
เวลา:    t=0      t=1      t=2      t=3    ...
สถานะ:   X₀  →    X₁  →    X₂  →    X₃    ...
         ↓        ↓        ↓        ↓
หลักฐาน:          E₁       E₂       E₃    ...
```

ส่วนประกอบของ Markov Model

เราต้องระบุ:

1. Prior distribution $\mathbf{P}(\mathbf{X}_0)$
    - กำหนดสถานะความเชื่อเริ่มต้นเกี่ยวกับตัวแปรสถานะที่ซ่อนอยู่
2. Transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1})$ (สำหรับ $t > 0$)
    - กำหนดการแจกแจงความน่าจะเป็นของตัวแปรสถานะล่าสุด จากค่าก่อนหน้า
3. Sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1})$ (สำหรับ $t > 0$)
    - กำหนดการแจกแจงความน่าจะเป็นของตัวแปรหลักฐานล่าสุด จากค่าที่ผ่านมาทั้งหมด

Markov Processes

สมมติฐาน Markov (Markov Assumption)

สถานะปัจจุบันของโลกขึ้นอยู่กับสถานะก่อนหน้าเพียงเล็กน้อยเท่านั้น กล่าวคือ $\mathbf{X}_t$ ขึ้นอยู่กับเซตย่อยที่จำกัดของ $\mathbf{X}_{0:t-1}$

กระบวนการสุ่มที่เป็นไปตามสมมติฐานนี้เรียกว่า Markov processes หรือ Markov chains

First-order Markov Processes

กระบวนการ Markov ที่มีคุณสมบัติ:

$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$

หมายความว่า $\mathbf{X}_t$ และ $\mathbf{X}_{0:t-2}$ เป็นอิสระเมื่อกำหนดค่า $\mathbf{X}_{t-1}$

Sensor Markov Assumption

เราสร้างสมมติฐาน sensor Markov assumption (อันดับหนึ่ง):

$$\mathbf{P}(\mathbf{E}_t | \mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t | \mathbf{X}_t)$$

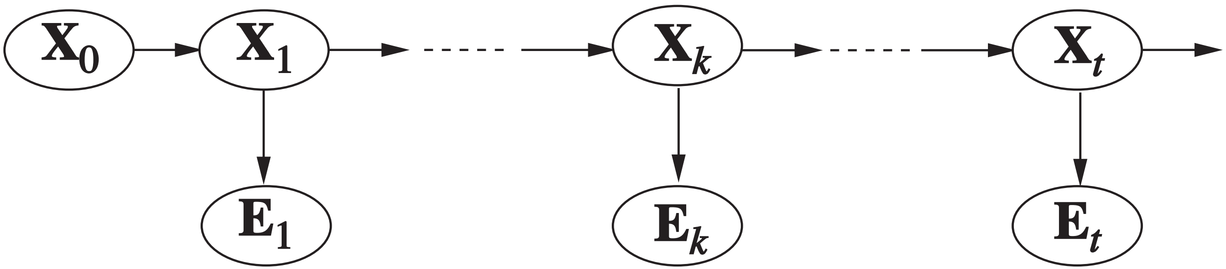
ความหมาย: ข้อมูลที่สังเกตได้ในปัจจุบันขึ้นอยู่กับสถานะปัจจุบันเท่านั้น

สมมติฐาน Stationarity

Transition model และ sensor model จะเหมือนกันสำหรับทุก t

กฎฟิสิกส์ไม่เปลี่ยนแปลงตามเวลา

การแจกแจงร่วม (Joint Distribution)

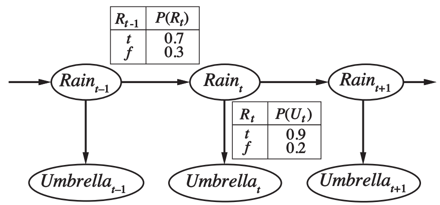


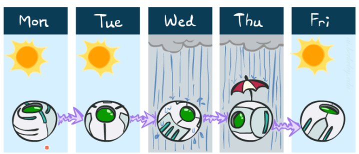
Markov chain ที่รวมกับ sensor model สามารถแสดงเป็น Bayesian network ที่ขยายได้ ซึ่งกางออกไปตามเวลาอนันต์

การแจกแจงร่วม ของตัวแปรทั้งหมดจนถึงเวลา t คือ:

$$\mathbf{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbf{P}(\mathbf{X}_0) \prod_{i=1}^{t} \mathbf{P}(\mathbf{X}_i | \mathbf{X}_{i-1}) \mathbf{P}(\mathbf{E}_i | \mathbf{X}_i)$$

ตัวอย่าง: วันนี้คุณจะเอาร่มไปหรือไม่?



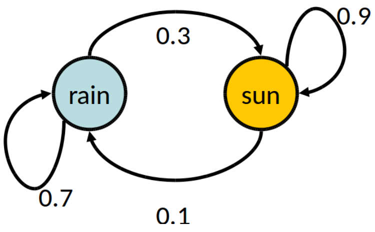


สถานการณ์:

- ฝนตก → เอาร่ม
- ไม่ฝนตก → ไม่เอาร่ม
- แต่เราไม่รู้ว่าจริงๆ ฝนตกหรือไม่

คำถามที่น่าสนใจ:

- $\mathbf{P}(\text{Umbrella}_t|\text{Rain}_t)$?
  - มีร่มเมื่อฝนตก?
- $\mathbf{P}(\text{Rain}_t|\text{Umbrella}_{0:t-1})$?
  - ฝนตกจากการเอาร่มก่อนหน้า?
- $\mathbf{P}(\text{Rain}_{t+2}|\text{Rain}_t)$?
  - ฝนตกในอนาคตจากปัจจุบัน?



Transition model $\mathbf{P}(\mathrm{Rain}_t|\mathrm{Rain}_{t-1})$ สามารถแสดงด้วย state transition diagram ได้

ตารางความน่าจะเป็นการเปลี่ยนผ่าน

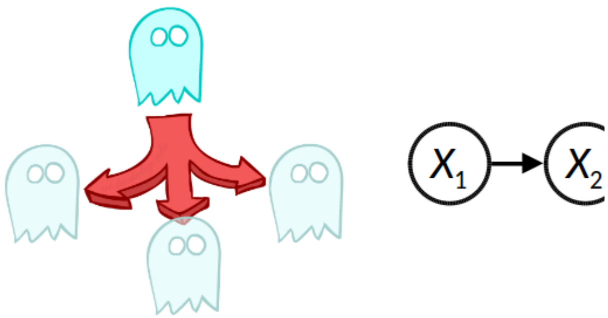
วันก่อน	**วันนี้**	**ความน่าจะเป็น**
ฝนตก	ฝนตก	0.7
ฝนตก	ไม่ฝน	0.3
ไม่ฝน	ฝนตก	0.2
ไม่ฝน	ไม่ฝน	0.8

งานการ Inference

ประเภทของการ Inference

1. Prediction (การทำนาย): $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ สำหรับ $k > 0$
    - คำนวณการแจกแจงความน่าจะเป็นของสถานะในอนาคต
    - ใช้สำหรับประเมินลำดับการกระทำที่เป็นไปได้
2. Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
    - การติดตามสถานะปัจจุบันที่ซ่อนอยู่ $\mathbf{X}_t$ (belief state)
    - สิ่งที่ rational agent ทำเพื่อตัดสินใจอย่างมีเหตุผล
3. Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ สำหรับ $0 \leq k < t$
    - คำนวณการแจกแจงความน่าจะเป็นของสถานะในอดีต
    - ใช้สร้างการประมาณที่ดีกว่า เนื่องจากรวมหลักฐานมากขึ้น
    - จำเป็นสำหรับการเรียนรู้
4. Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
    - ใช้ในการถอดรหัส (decoding) ช่องทางสัญญาณรบกวน, การรู้จำเสียงพูด ฯลฯ

กรณีฐาน (Base Cases)



การทำนาย (Prediction)

$$\mathbf{P}(\mathbf{X}_2) = \sum_{\mathbf{x}_1} \mathbf{P}(\mathbf{X}_2, \mathbf{x}_1) = \sum_{\mathbf{x}_1} P(\mathbf{x}_1)\mathbf{P}(\mathbf{X}_2|\mathbf{x}_1)$$

ขั้นตอน: ส่ง $\mathbf{P}(\mathbf{X}_1)$ ข้างหน้าผ่าน transition model

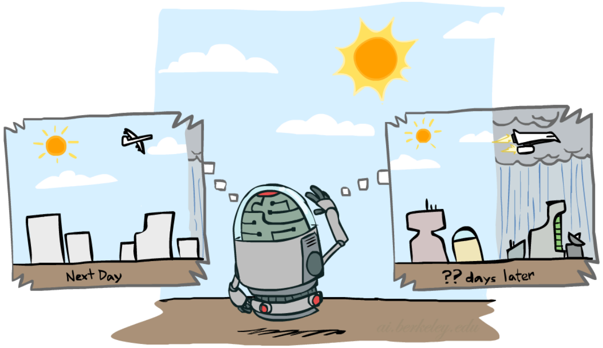
การปรับปรุง (Update)

$$\mathbf{P}(\mathbf{X}_1|\mathbf{e}_1) = \frac{\mathbf{P}(\mathbf{e}_1|\mathbf{X}_1)\mathbf{P}(\mathbf{X}_1)}{P(\mathbf{e}_1)} \propto \mathbf{P}(\mathbf{e}_1|\mathbf{X}_1)\mathbf{P}(\mathbf{X}_1)$$

ขั้นตอน: ปรับปรุง $\mathbf{P}(\mathbf{X}_1)$ ด้วยหลักฐาน $\mathbf{e}_1$ โดยใช้ sensor model

—
เครดิต: CS188, UC Berkeley.

การทำนาย (Prediction)



เพื่อทำนายอนาคต $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$:

ขั้นตอนการทำนาย

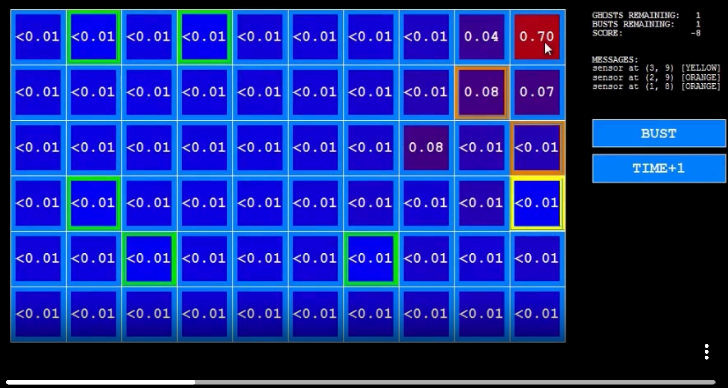
1. ส่งต่อ (Push) สถานะความเชื่อก่อนหน้า $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ ผ่าน transition model:

$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) = \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t})$

1. ทำซ้ำ จนถึง $t+k$ โดยใช้ $\mathbf{P}(\mathbf{X}_{t+k-1}|\mathbf{e}_{1:t})$ ในการคำนวณ $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$

หลักการ

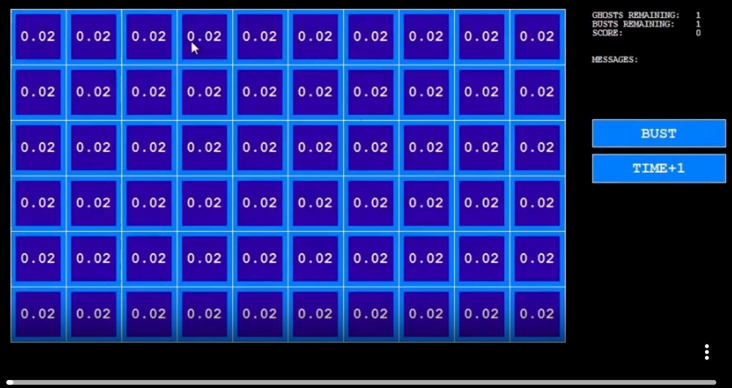
- ยิ่งไกลในอนาคต → ความไม่แน่นอนมากขึ้น
- ไม่มีหลักฐานใหม่ → การแจกแจงกระจายออก



พลศาสตร์แบบสุ่ม (Random Dynamics)

การเคลื่อนที่แบบไม่มีทิศทาง

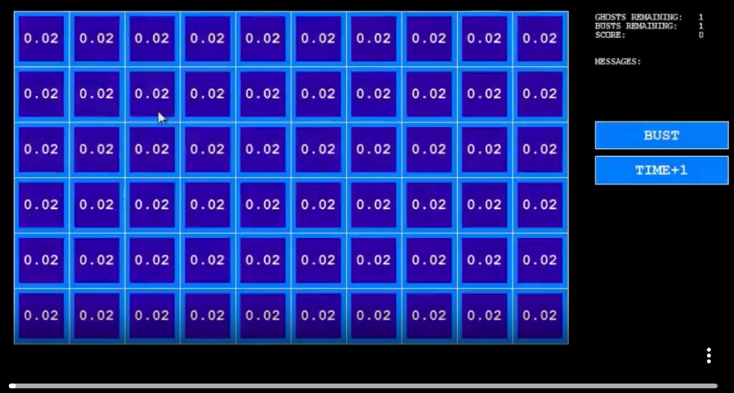
—
เครดิต: CS188, UC Berkeley.



พลศาสตร์แบบวงกลม (Circular Dynamics)

การเคลื่อนที่เป็นวงกลม

เครดิต: CS188, UC Berkeley.

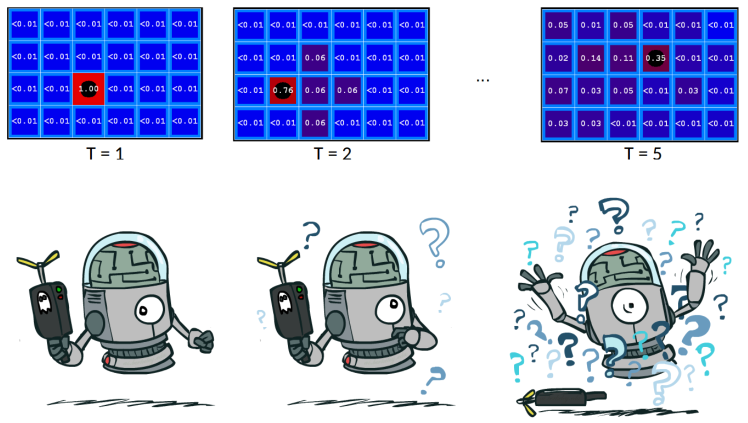


พลศาสตร์แบบน้ำวน (Whirlpool Dynamics)

การเคลื่อนที่เป็นเกลียวหมุน

เครดิต: CS188, UC Berkeley.

ความไม่แน่นอนเพิ่มขึ้นตามเวลา



สิ่งสำคัญที่ต้องจำ

- เมื่อเวลาผ่านไป → ความไม่แน่นอน (โดยทั่วไป) เพิ่มขึ้น
- ไม่มีหลักฐานใหม่ → การแจกแจงกระจายออก
- มีหลักฐานใหม่ → ความแน่นอนเพิ่มขึ้น

—
เครดิต: CS188, UC Berkeley.

การแจกแจงคงที่ (Stationary Distributions)

จะเกิดอะไรขึ้นเมื่อ $t \to \infty$?

- สำหรับ chain ส่วนใหญ่ อิทธิพลของการแจกแจงเริ่มต้นจะลดลงเรื่อยๆ ตามเวลา
- ในที่สุด การแจกแจงจะเข้าสู่จุดคงที่ เรียกว่า stationary distribution
- การแจกแจงนี้มีคุณสมบัติ:

$$\mathbf{P}(\mathbf{X}_\infty) = \mathbf{P}(\mathbf{X}_{\infty+1}) = \sum_{\mathbf{x}_\infty} \mathbf{P}(\mathbf{X}_{\infty+1}|\mathbf{x}_\infty)P(\mathbf{x}_\infty)$$

ความหมาย

- การแจกแจงไม่เปลี่ยนแปลงอีกต่อไป
- เป็นสมดุลระยะยาว (Long-term equilibrium)
- ไม่ขึ้นกับสถานะเริ่มต้น

ตัวอย่างการคำนวณ Stationary Distribution

ตารางการเปลี่ยนผ่านสภาพอากาศ

$\mathbf{X}_{t-1}$	$\mathbf{X}_t$	P
แดด	แดด	0.9
แดด	ฝน	0.1
ฝน	แดด	0.3
ฝน	ฝน	0.7

การแก้สมการ

$$\begin{aligned} P(\mathbf{X}_\infty = \text{แดด}) &= P(\mathbf{X}_{\infty+1} = \text{แดด}) \\ &= P(\mathbf{X}_{\infty+1} = \text{แดด} | \mathbf{X}_\infty = \text{แดด}) P(\mathbf{X}_\infty = \text{แดด}) \\ &\quad + P(\mathbf{X}_{\infty+1} = \text{แดด} | \mathbf{X}_\infty = \text{ฝน}) P(\mathbf{X}_\infty = \text{ฝน}) \\ &= 0.9 P(\mathbf{X}_\infty = \text{แดด}) + 0.3 P(\mathbf{X}_\infty = \text{ฝน}) \end{aligned}$$

ผลลัพธ์: $P(\mathbf{X}_\infty = \text{แดด}) = 3P(\mathbf{X}_\infty = \text{ฝน})$

เนื่องจาก $P(\text{แดด}) + P(\text{ฝน}) = 1$

- $P(\mathbf{X}_\infty = \text{แดด}) = 0.75$
- $P(\mathbf{X}_\infty = \text{ฝน}) = 0.25$

การ Filtering

การติดตามสถานะปัจจุบัน

อัลกอริทึม Forward Algorithm

การคำนวณ $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$ แบบเวียนเกิด (recursive):

ขั้นตอน Predict (ทำนาย)

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) = \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{e}_{1:t})$$

ขั้นตอน Update (ปรับปรุง)

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) \propto \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})$$

ลำดับการทำงาน

1. เริ่มจาก $\mathbf{P}(\mathbf{X}_0)$ (prior)
2. สำหรับแต่ละ t:
   - Predict: คำนวณ $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t-1})$
   - Update: คำนวณ $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$

ตัวอย่างการ Filtering: Ghost Tracking

สถานการณ์

- Agent (Pacman) อยู่ที่ตำแหน่งที่ทราบ
- Ghost อยู่ที่ตำแหน่งที่ไม่ทราบ (สถานะที่ซ่อน)
- มี sensor วัดระยะทางแต่ไม่แม่นยำ (noisy distance reading)

โจทย์

ติดตามตำแหน่งของ Ghost ตามเวลาโดยใช้ sensor readings

Hidden Markov Models (HMMs)

แบบจำลองที่มีสถานะซ่อน

Hidden Markov Models

Hidden Markov Model คือ Markov chain ที่:

- สถานะ $\mathbf{X}_t$ ไม่สามารถสังเกตได้โดยตรง (hidden)
- การสังเกต $\mathbf{E}_t$ สามารถมองเห็นได้ และขึ้นอยู่กับสถานะ

องค์ประกอบของ HMM

1. States $\mathbf{X}_t \in \{s_1, s_2, ..., s_n\}$
   - สถานะที่ซ่อนอยู่ n สถานะ
2. Observations $\mathbf{E}_t \in \{o_1, o_2, ..., o_m\}$
   - การสังเกตที่มองเห็นได้ m รูปแบบ
3. Transition probabilities $A_{ij} = P(X_{t+1} = s_j | X_t = s_i)$
   - ความน่าจะเป็นการเปลี่ยนสถานะ

Hidden Markov Models

Hidden Markov Model คือ Markov chain ที่:

- สถานะ $\mathbf{X}_t$ ไม่สามารถสังเกตได้โดยตรง (hidden)
- การสังเกต $\mathbf{E}_t$ สามารถมองเห็นได้ และขึ้นอยู่กับสถานะ

องค์ประกอบของ HMM (ต่อ)

1. Emission probabilities $B_{jk} = P(E_t = o_k | X_t = s_j)$
    - ความน่าจะเป็นการสังเกตในแต่ละสถานะ
2. Initial probabilities $\pi_i = P(X_0 = s_i)$
    - ความน่าจะเป็นเริ่มต้น

ตัวอย่าง HMM: การวินิจฉัยทางการแพทย์

สถานการณ์

- สถานะที่ซ่อน: สุขภาพ (ดี/ไม่ดี)
- การสังเกต: อาการ (มีไข้/ไม่มีไข้, ปวดหัว/ไม่ปวดหัว)

Transition Matrix

	ดี	**ไม่ดี**
ดี	0.7	0.3
ไม่ดี	0.4	0.6

Initial State

- P(ดี) = 0.6
- P(ไม่ดี) = 0.4

Emission Matrix

	ไม่มีอาการ	**มีอาการ**
ดี	0.8	0.2
ไม่ดี	0.1	0.9

คำถาม

ถ้าผู้ป่วยมีอาการ 3 วันติดต่อกัน ความน่าจะเป็นที่สุขภาพไม่ดีแต่ละวันคือเท่าไร?

การประยุกต์ใช้ HMMs

1. การรู้จำเสียงพูด (Speech Recognition)

- สถานะ: phonemes (หน่วยเสียง)
- การสังเกต: acoustic features

2. การรู้จำลายมือ (Handwriting Recognition)

- สถานะ: letters/words
- การสังเกต: pen strokes/pixels

3. Bioinformatics

- สถานะ: gene structure (exon/intron)
- การสังเกต: DNA sequences

4. การวิเคราะห์ตลาดการเงิน

- สถานะ: market regimes (bull/bear)
- การสังเกต: stock prices/returns

Filters (ตัวกรอง)

เครื่องมือสำหรับการติดตามสถานะ

Kalman Filter

Kalman Filter เป็น optimal filter สำหรับระบบ linear Gaussian:

สมมติฐาน

- สถานะ: ตัวแปรต่อเนื่อง (continuous variables)
- Transition model: linear transformation + Gaussian noise
- Sensor model: linear transformation + Gaussian noise

สมการ Kalman Filter

Predict step (การทำนาย):

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t$$

โดยที่:

- $\mathbf{F}_t$ = transition matrix
- $\mathbf{Q}_t$ = process noise covariance

Update step (การปรับปรุง):

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1}$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1})$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}$$

โดยที่:

- $\mathbf{H}_t$ = observation matrix
- $\mathbf{R}_t$ = measurement noise covariance

ตัวอย่าง Kalman Filter: การติดตามวัตถุ

สถานการณ์: ติดตามรถยนต์ด้วย GPS

สถานะ (State):

- ตำแหน่ง (x, y)
- ความเร็ว (v_x, v_y)

การสังเกต (Observation):

- GPS readings (มี noise)
- ความเร็วจาก speedometer

ข้อดีของ Kalman Filter

- Optimal สำหรับระบบ linear Gaussian
- คำนวณเร็ว (complexity เป็น $O(n^3)$)
- ใช้หน่วยความจำน้อย (เก็บแค่ state และ covariance)

แบบจำลอง:

$$\mathbf{x}\_t = [x\ y\ v\_x\ v\_y]\ \_t$$

Transition matrix:

$$\mathbf{F} = [1 \quad 0 \quad \Delta t \quad 0\, 0 \quad 1 \quad 0 \quad \Delta t\, 0 \quad 0 \quad 1 \quad 0\, 0 \quad 0 \quad 0 \quad 1]$$

Particle Filter

Particle Filter ใช้สำหรับกรณีที่ไม่ใช่ linear/Gaussian:

หลักการ

- ใช้ particles (samples) เป็นตัวแทนการแจกแจงความน่าจะเป็น
- แต่ละ particle คือสถานะที่เป็นไปได้หนึ่งสถานะ
- จำนวน particles ในแต่ละพื้นที่แสดงความน่าจะเป็น

ขั้นตอนการทำงาน

1. Initialization: สร้าง N particles จาก prior distribution
2. Prediction: เลื่อน particles ตาม transition model

```
สำหรับแต่ละ particle i:
 x_t^(i) = sample จาก P(X_t | x_{t-1}^(i))
```

3. Update: คำนวณ weight จากความน่าจะเป็นของการสังเกต

```
สำหรับแต่ละ particle i:
 w_t^(i) = P(e_t | x_t^(i))
```

4. Resampling: เลือก particles ใหม่ตาม weights

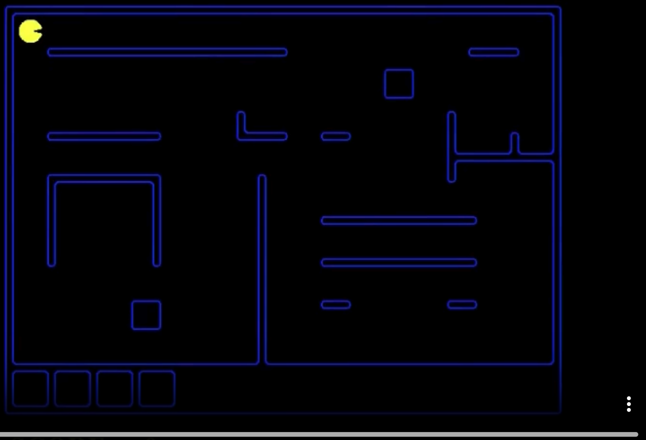
ตัวอย่าง Particle Filter: Robot Localization

สถานการณ์

- Robot เคลื่อนที่ในพื้นที่ที่ทราบแผนที่
- Sensors: วัดระยะทางถึงผนัง (มี noise)
- โจทย์: หาตำแหน่งของ robot

การแสดงผล

- จุดสีแดง: particles (สถานะที่เป็นไปได้)
- ความหนาแน่น: แสดงความน่าจะเป็น
- เส้นสีเขียว: sensor readings



Pacman ด้วย Particle Filter

การติดตามตำแหน่งผี (ghost) ด้วย belief state

—

เครดิต: CS188, UC Berkeley.

```
python3 run.py --nghosts 2 --layout maze_small --agentfile sherlockpacman.py --bsagentfile bayesfilter.py --show True
python3 run.py --nghosts 3 --layout maze_medium --agentfile sherlockpacman.py --bsagentfile bayesfilter.py --show True
python3 run.py --nghosts 4 --layout maze_huge --agentfile sherlockpacman.py --bsagentfile bayesfilter.py --show True
```

การเปรียบเทียบ Filters

	Kalman Filter	Particle Filter
ประเภทสถานะ	Continuous	Discrete/Continuous
Transition Model	Linear + Gaussian	ใดก็ได้
Sensor Model	Linear + Gaussian	ใดก็ได้
การแสดง Belief	Gaussian (μ, Σ)	Set of particles
ความแม่นยำ	Optimal (หากสมมติฐานถูก)	Approximate
ความเร็ว	เร็วมาก $O(n^3)$	ช้ากว่า $O(N \times n)$
หน่วยความจำ	น้อย $O(n^2)$	มาก $O(N \times n)$
ความยืดหยุ่น	จำกัด	สูงมาก

เมื่อไหร่ใช้อะไร?

- Kalman: ระบบ linear, Gaussian, ต้องการความเร็ว
- Particle: ระบบ nonlinear, non-Gaussian, ต้องการความยืดหยุ่น

การประยุกต์ใช้จริง

ตัวอย่างการใช้งานในโลกแห่งความเป็นจริง

การประยุกต์ใช้ในชีวิตจริง

1. ระบบนำทาง GPS

- Kalman Filter สำหรับติดตามตำแหน่งรถ
- รวม GPS, accelerometer, gyroscope
- ประมาณความเร็วและทิศทาง

2. รถยนต์ไร้คนขับ (Autonomous Vehicles)

- Particle Filter สำหรับ localization
- Kalman Filter สำหรับติดตามวัตถุ
- รวมข้อมูล camera, LiDAR, radar

3. หุ่นยนต์ในโรงงาน

- ติดตามตำแหน่งชิ้นงาน
- วางแผนเส้นทางการเคลื่อนที่
- หลีกเลี่ยงสิ่งกีดขวาง

4. การตรวจจับใบหน้า

- HMM สำหรับติดตาม facial landmarks
- Particle Filter สำหรับติดตามการเคลื่อนไหว
- การรู้จำอารมณ์และการแสดงออก

5. การวิเคราะห์การเงิน

- HMM สำหรับหา market regimes
- Kalman Filter สำหรับ portfolio optimization
- การทำนายราคาหุ้น

6. เกมและความบันเทิง

- Particle Filter ใน game AI
- การติดตามผู้เล่นใน augmented reality
- การสร้าง intelligent NPCs

สรุปและข้อคิด

สรุปสำคัญ

แนวคิดหลัก

1. Markov Assumption: อนาคตขึ้นอยู่กับปัจจุบัน ไม่ใช่อดีต
2. Belief State: การรักษาความน่าจะเป็นของสถานะที่เป็นไปได้
3. Predict-Update Cycle: วงจรการทำนายและปรับปรุง

เครื่องมือสำคัญ

- Hidden Markov Models: สำหรับระบบ discrete states
- Kalman Filter: สำหรับระบบ linear Gaussian
- Particle Filter: สำหรับระบบทั่วไป non-linear/non-Gaussian

การประยุกต์

- การติดตามวัตถุ (Object tracking)
- การระบุตำแหน่ง (Localization)
- การรู้จำรูปแบบ (Pattern recognition)
- การควบคุมระบบ (Control systems)

คำถามสำหรับคิด

1. ทำไม Markov assumption ถึงมีประโยชน์?
    - ลดความซับซ้อน computational
    - ทำให้สามารถแก้ปัญหาได้
2. เมื่อไหร่ ควรใช้ Particle Filter แทน Kalman Filter?
    - ระบบ non-linear
    - การแจกแจงไม่ใช่ Gaussian
    - ต้องการความยืดหยุ่น
3. อย่างไร จะปรับปรุงประสิทธิภาพ?
    - เพิ่มจำนวน particles
    - ปรับปรุง resampling algorithm
    - ใช้ adaptive techniques

การเชื่อมโยงกับบทบรรยายอื่นๆ

บทก่อนหน้า

- Probability & Bayes' Theorem → พื้นฐานการคำนวณ
- Bayesian Networks → โครงสร้างการแทนความรู้

บทต่อไป

- Decision Making → การใช้ belief state ในการตัดสินใจ
- Reinforcement Learning → การเรียนรู้ในสภาพแวดล้อมที่ไม่แน่นอน
- Planning under Uncertainty → การวางแผนเมื่อไม่ทราบสถานะแน่นอน

แบบฝึกหัดและโปรเจ็กต์

แบบฝึกหัดที่แนะนำ

1. การคำนวณ Stationary Distribution

จากตาราง transition probabilities ให้หา stationary distribution

2. การ Implement Simple Particle Filter

สร้าง particle filter สำหรับปัญหา 1D tracking

3. การวิเคราะห์ HMM

จากข้อมูลการสังเกต ให้หาความน่าจะเป็นของสถานะที่ซ่อน

4. การออกแบบ Sensor Model

ออกแบบ sensor model สำหรับปัญหา robot localization

5. การเปรียบเทียบ Filters

เปรียบเทียบประสิทธิภาพ Kalman vs Particle Filter

คำถาม - ข้อสงสัย - อภิปราย

บทบรรยายหน้า: การตัดสินใจภายใต้ความไม่แน่นอน
(Decision Making under Uncertainty)

เอกสารอ้างอิง

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th Edition.
2. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics. MIT Press.
3. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer.
4. Kevin Murphy. Machine Learning: A Probabilistic Perspective. MIT Press.
5. Daphne Koller and Nir Friedman. Probabilistic Graphical Models. MIT Press.

แหล่งข้อมูลออนไลน์

- CS188 UC Berkeley: https://inst.eecs.berkeley.edu/~cs188/
- Stanford CS221: https://stanford-cs221.github.io/

—
เครดิตต้นฉบับ: CS188, UC Berkeley.