11/9 $\boxed{\text{Review}}$

Gaussian mixture model

* Belief network

$$Z \longrightarrow \vec{x}$$

$P(Z=i)=\pi_i \qquad P(\vec{x}\,|\,Z=i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\, e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)}$

* ML estimation for complete data $\{(z_t, \vec{x}_t)\}_{t=1}^{T}$

$$\pi_i = T_i/T \quad \text{where } T_i = \sum_{t=1}^{T} I(z_t, i)$$

$$\vec{\mu}_i = \frac{1}{T_i}\sum_{t=1}^{T}\vec{x}_t\, I(z_t, i)$$

$$\Sigma_i = \frac{1}{T_i}\sum_{t=1}^{T}(\vec{x}_t-\vec{\mu}_i)(\vec{x}_t-\vec{\mu}_i)^T I(z_t, i)$$

* EM algorithm for incomplete data.

- E-step: Compute posterior probability:

Gaussian

π_i

$$P(Z=i\,|\,\vec{x}_t) = \frac{P(\vec{x}_t\,|\,Z=i)\,P(Z=i)}{\sum_{j=1}^{K}P(\vec{x}_t\,|\,Z=j)\,P(Z=j)}$$

- M-step: update CPTs (by analogy to complete data case).

$$\pi_i \longleftarrow \frac{1}{T}\sum_t P(Z=i\,|\,\vec{x}_t)$$

$$\vec{\mu}_i \longleftarrow \frac{\sum_t \vec{x}_t\, P(Z=i\,|\,\vec{x}_t)}{\sum_t P(Z=i\,|\,\vec{x}_t)}$$

$\sum_t P(Z=i\,|\,\vec{x}_t)$ ← effective # points assigned to i^{th} cluster.

$$\Sigma_i \longleftarrow \frac{\sum_t(\vec{x}_t-\vec{\mu}_i)(\vec{x}_t-\vec{\mu}_i)^T P(Z=i\,|\,\vec{x}_t)}{\sum_t P(Z=i\,|\,\vec{x}_t)}$$

from previous update.

$\boxed{\text{Linear dynamical systems}}$

HMM = discrete dynamical system

How to model continuous states and observations?

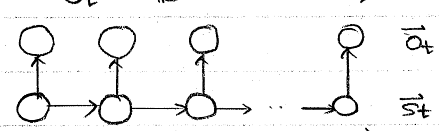
Ex: missile locations and radar measurements

* Belief network

variables: $\vec{s}_t \in \mathbb{R}^n$ (hidden)

$\vec{o}_t \in \mathbb{R}^m$ (observed)

DAG:



CPTs: $P(\vec{s}_1) = N(\vec{s}_1;\ \vec{\mu}_1, \Sigma_1)$

$P(\vec{s}_{t+1}\,|\,\vec{s}_t) = N(\vec{s}_{t+1};\ A\vec{s}_t, \Sigma_H)$

$P(\vec{o}_t\,|\,\vec{s}_t) = N(\vec{o}_t;\ B\vec{s}_t, \Sigma_O)$

n×n matrix parameter

n-dim vector.

n×n matrix

m×n matrix

m×m matrix

* Belief updating

What is $P(\vec{s}_t \mid \vec{o}_1, \vec{o}_2, \cdots, \vec{o}_t)$? "Kalman filtering".

Recall key property : in a BN with all Gaussian CPTs, every marginal and posterior probability is also Gaussian.

Hence $P(\vec{s}_t \mid \vec{o}_1, \cdots, \vec{o}_t)$ must also be gaussian.

Enough to track $\vec{\mu}_t$ and Σ_t.

Recursion (stated w/o proof) : "Kalman gain" matrix corrects for errors

$$\vec{\mu}_{t+1} = \underbrace{A\vec{\mu}_t}_{\text{predicted state}} + K_{t+1}\underbrace{(O_{t+1} - BA\vec{\mu}_t)}_{\text{error of predicted observation.}}$$

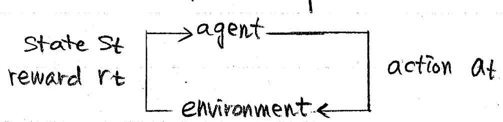
predicted state from evolution of $\vec{\mu}_t$ at time t

mean of $P(S_{t+1} \mid S_t = \vec{\mu}_t)$

Quiz #2

Reinforcement learning

Q : How should embodied / embedded / interactive decision-making agents learn from experience in the world?



Ex : robot navigation,
chess / backgammon,
elevator scheduling

* Challenges

- handling uncertainty
- exploration vs. exploitation dilemma.
- temporal credit assignment : delayed vs. immediate rewards.
- evaluative vs. instructive feedback.
- complex worlds : balance representational power vs. tractability
  computational guarantees : convergence, optimality, efficiency, etc ...

Markov decision process (MDP)

* Definition

- state space $\mathcal{S}$ with states $s \in \mathcal{S}$
- action space $\mathcal{A}$ with actions $a \in \mathcal{A}$
- transition probabilities for all state-action pairs (s, a)

$P(s' \mid s, a) \triangleq P(S_{t+1} = s' \mid s_t = s, a_t = a)$: probability of moving from state s to state s' given action a

* Assumptions

- time independent : $P(s_{t+1} = s' \mid s_t = s, a_t = a) = P(s_t = s' \mid s_{t-1} = s, a_{t-1} = a)$
- conditional independence : $P(s_{t+1} \mid s_t, a_t) = P(s_{t+1} \mid s_t, a_t, s_{t-1}, a_{t-1}, \cdots, s_0, a_0)$

* Reward function

$R(s, s', a)$ = real-valued reward after taking action a in state s and moving to state s'.

$$\boxed{MDP = \{\mathcal{S}, \mathcal{A}, P(s' \mid s, a), R(s, s', a)\}}$$

* Simplifications (for CSE 250A)

- Reward function $R(s, s', a) = R(s) = R_s$ (only depends on current state)
- Bounded rewards $\max_s |R_s| < \infty$.
- Deterministic rewards
- Discrete & finite state space $\mathcal{S}$
- " " " action space $\mathcal{A}$

* Ex : backgammon

$\mathcal{S}$ = board position & roll of dice

$\mathcal{A}$ = set of possible moves

$P(s' \mid s, a)$ = agent's move, opponent rolls dice, opponent's move, agent rolls dice.

$$R(s) = \begin{cases} +1 & \text{win} \\ -1 & \text{lose} \\ 0 & \text{otherwise.} \end{cases}$$

* Decision-making

- policy : deterministic mapping $\pi : \mathcal{S} \to \mathcal{A}$ from states to actions
- # policies $= |\mathcal{A}|^{|\mathcal{S}|}$
- dynamics $P(s' \mid s, \pi(s))$
- experience

$$\begin{array}{lccccccc} \text{state} & s_0 & \xrightarrow{\pi(s_0)} & s_1 & \xrightarrow{\pi(s_1)} & s_2 & \longrightarrow & \cdots \\ \text{reward} & r_0 & & r_1 & & r_2 & & \end{array}$$

* How to measure long-term return (accumulated rewards)?

- return $= \frac{1}{T}[r_0 + r_1 + r_2 + \cdots + r_{T-1}]$

  undiscounted return w/ finite horizon T

- return $= \lim_{T\to\infty} \frac{1}{T}[r_0 + r_1 + \cdots + r_{T-1}]$

  undiscounted infinite horizon

- discounted infinite horizon return.

  discount factor $0 \le \gamma < 1$. return $= r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots = \sum_{t=0}^{\infty} \gamma^t r_t$

possibilities: $\gamma = 0 \rightarrow$ only immediate reward matters

$\gamma \ll 1 \rightarrow$ near-sighted agent

$(1-\gamma) \ll 1 \rightarrow$ far-sighted agent

* Justification

intuitive: near future weighted more than distant future

mathematically convenient: leads to recursive algorithms

* State value function (over discounted infinite horizon)

$V^{\pi}(s)$ = expected long term return following policy π from initial state s.

$$= E^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s \right]$$

- Maximizing expected return different than:
  - maximizing worst-case return
  - maximizing best-case return.