Gaussian mixture model

* Belief network

\[ \mathbf{Z} \rightarrow \mathbf{X} \]

\[ P(\mathbf{Z}=i | \mathbf{X}) = \frac{1}{\pi^k |\Sigma_i|^{1/2}} e^{-\frac{1}{2} (\mathbf{X} - \mu_i)^T \Sigma_i^{-1} (\mathbf{X} - \mu_i)} \]

* ML estimation for complete data \( \{ (Z_t, X_t) \}_{t=1}^T \)

\[ \pi_i = \frac{T_i}{T} \quad \text{where} \quad T_i = \sum_{t=1}^T I(Z_t = i) \]

\[ \mu_i = \frac{1}{T_i} \sum_{t=1}^T X_t \cdot I(Z_t = i) \]

\[ \Sigma_i = \frac{1}{T_i} \sum_{t=1}^T (X_t - \mu_i)(X_t - \mu_i)^T \cdot I(Z_t = i) \]

* EM algorithm for incomplete data

- E-step: Compute posterior probability of Gaussian

\[ P(\mathbf{Z}=i | \mathbf{X}_t) = \frac{P(\mathbf{X}_t | \mathbf{Z}=i) \cdot P(\mathbf{Z}=i)}{\sum_{j=1}^k P(\mathbf{X}_t | \mathbf{Z}=j) \cdot P(\mathbf{Z}=j)} \]

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

\[ \pi_i \leftarrow \frac{1}{T} \sum_{t=1}^T P(\mathbf{Z}=i | \mathbf{X}_t) \]

\[ \mu_i \leftarrow \frac{\sum_{t=1}^T \mathbf{X}_t \cdot P(\mathbf{Z}=i | \mathbf{X}_t)}{\sum_{t=1}^T P(\mathbf{Z}=i | \mathbf{X}_t)} \quad \text{effective # points assigned to } i^{th} \text{ cluster} \]

\[ \Sigma_i \leftarrow \frac{\sum_{t=1}^T (\mathbf{X}_t - \mu_i)(\mathbf{X}_t - \mu_i)^T \cdot P(\mathbf{Z}=i | \mathbf{X}_t)}{\sum_{t=1}^T P(\mathbf{Z}=i | \mathbf{X}_t)} \quad \text{from previous update} \]

Linear dynamical systems

HMM = discrete dynamical system

How to model continuous states and observations?

Ex: missile locations and radar measurements

* Belief network

variables: \( \mathbf{S}_t \in \mathbb{R}^n \) (hidden)

\( \mathbf{O}_t \in \mathbb{R}^m \) (observed)

DAG:

\[ \circ \rightarrow \circ \rightarrow \circ \rightarrow \cdots \rightarrow \circ \rightarrow \circ \]

CPTs:

\[ P(\mathbf{S}_t | \mathbf{S}_{t-1}) = \mathcal{N}(\mathbf{S}_t; \mu_t, \Sigma_t) \]

\[ P(\mathbf{O}_t | \mathbf{S}_t) = \mathcal{N}(\mathbf{O}_t; \mathbf{A}\mathbf{S}_t, \Sigma_H) \]

\[ P(\mathbf{S}_t | \mathbf{O}_t) = \mathcal{N}(\mathbf{S}_t; \mathbf{B}\mathbf{O}_t, \Sigma_0) \]
Belief updating

What is \( P(\mathbf{\hat{X}} | \mathbf{\hat{O}}_1, \mathbf{\hat{O}}_2, \ldots, \mathbf{\hat{O}}_t) \)? "Kalman filtering".

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

Hence \( P(\mathbf{\hat{X}} | \mathbf{\hat{O}}_1, \ldots, \mathbf{\hat{O}}_t) \) must also be Gaussian.

Enough to track \( \mathbf{\hat{M}}_t \) and \( \mathbf{\Sigma}_t \).

Recursion (stated w/o proof):

\[
\mathbf{\hat{M}}_{t+1} = \mathbf{\overline{M}}_t + \mathbf{K}_{t+1}(\mathbf{\hat{O}}_{t+1} - \mathbf{B} \mathbf{\hat{M}}_t)
\]

Kalman gain matrix corrects for errors from evolution of \( \mathbf{\hat{M}}_t \) at time \( t \) mean of \( P(\mathbf{\hat{M}}_t | \mathbf{\hat{S}}_t = \mathbf{\overline{M}}_t) \)

Reinforcement learning

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

state \( \mathbf{S}_t \)

agent

reward \( \mathbf{R}_t \)

environment

action \( \mathbf{A}_t \)

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

\[
P(\mathbf{s}' | s, a) = P(\mathbf{s}_{t+1} = \mathbf{s}' | \mathbf{s}_t = s, \mathbf{a}_t = a) \text{ probability of moving from state } s \text{ to state } \mathbf{s}' \text{ given action } a\]
* Assumptions
  - Time independent: \( P(S_{t+1} = S' \mid S_t = S, a_t = a) = P(S_{t+1} = S, a_{t+1} = a) \)
  - Conditional independence: \( P(S_{t+1} \mid S_t, a_t) = P(S_{t+1} \mid S_t, a_t, S_{t-1}, a_{t-1}, \ldots, S_0, a_0) \)
* Reward function
  \( R(s, s', a) = \text{real-valued reward after taking action } a \text{ in state } s \text{ and moving to state } s' \)

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

* 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
  \( \mathcal{S} = \text{board position & roll of dice} \)
  \( \mathcal{A} = \text{set of possible moves} \)
  \( P(s' \mid s, a) = \text{agent's move, opponent rolls dice, opponent's move, agent rolls dice} \)
  \( R(s) = \begin{cases} 
    f+1 & \text{win} \\
    -1 & \text{lose} \\
    0 & \text{otherwise} 
  \end{cases} \)

* Decision-making
  - Policy: deterministic mapping \( \pi: \mathcal{S} \rightarrow \mathcal{A} \) from states to actions
  - # Policies = \( |\mathcal{A}|^{|\mathcal{S}|} \)
  - Dynamics \( P(s' \mid s, \pi(s)) \)
  - Experience state sequence \( s_0 \xrightarrow{\pi(s_0)} s_1 \xrightarrow{\pi(s_1)} s_2 \rightarrow \cdots \)

* How to measure long-term return (accumulated rewards)?
  - Return = \( \frac{1}{1 - \gamma} \left[ r_0 + r_1 + r_2 + \cdots + r_{T-1} \right] \)
    - undiscounted return w/ finite horizon \( T \)
  - Return = \( \lim_{T \to \infty} \frac{1}{1 - \gamma} \left[ r_0 + r_1 + \cdots + r_{T-1} \right] \)
    - undiscounted infinite horizon
  - Discounted infinite horizon return
    discount factor \( 0 \leq \gamma < 1 \)
    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
\( \gamma < 1 \rightarrow \) near-sighted agent
\( (1-\gamma) < 1 \rightarrow \) 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)*
\[ V^\pi(s) = \text{expected long term return following policy } \pi \text{ from initial state } s \]
\[ = E^\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s \right] \]
- Maximizing expected return different than:
  - maximizing worst-case return
  - maximizing best-case return