CS 486/686
Hidden Markov Models I

Yuntian Deng

Lecture 10

RN 14.1 & 14.2.1 · PM 8.5.1–8.5.3

Search › **Uncertainty** › Decisions › Learning

Learning goals

- **Construct** a hidden Markov model from a real-world scenario.
- State the **independence assumptions** baked into an HMM.
- Compute the **filtering probability** for a time step.
- **Justify** each step in the derivation of the filtering formula.

Inference in a changing world

So far, we reasoned about a *static* world. But the world evolves over time — we need to reason about **sequences of events**.

Weather

Stocks

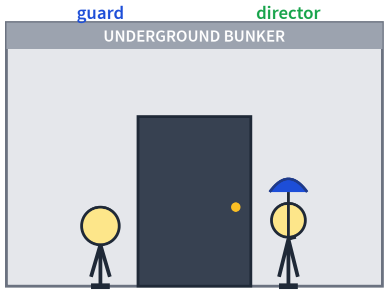
Patient monitor

Robot localization

Speech

Today's running example: an **umbrella story** — the simplest possible HMM.

The umbrella story



You are a **security guard** stationed at a secret underground facility.

You want to know whether it is **raining outside** today.

Your only signal: each morning, the director walks in **with** or **without** an umbrella.

States and observations

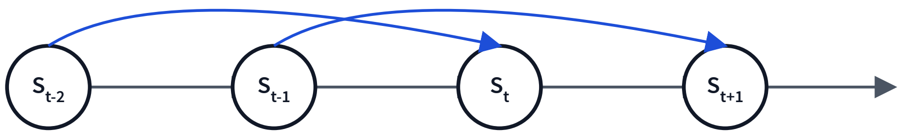
- The world contains a series of **time slices**.
- Each slice has a hidden **state** S_t and a visible **observation** O_t.

Q. What are the random variables in the umbrella world?

- S_t = *true* iff it rains at time t (hidden).
- O_t = *true* iff the director carries an umbrella at time t (observed).

The Markov assumption

"The future is independent of the past given the present."



First-order Markov (the default):

$$P(S_t|S_{t-1},\ldots,S_0) = P(S_t|S_{t-1})$$

Second-order: also depends on S_{t-2} (blue skip-edges).

$$P(S_t|S_{t-1},S_{t-2})$$

Stationary process: the same CPT is used at every time step.

Transition model

Transition CPT

$P(s_0) = 0.5$
$P(s_t | s_{t-1}) = 0.7$
$P(s_t | \neg s_{t-1}) = 0.3$

Warm-up: what is $P(S_K)$ for large K?

$$P(S_1 = T) = 0.5 \cdot 0.7 + 0.5 \cdot 0.3 = 0.5$$

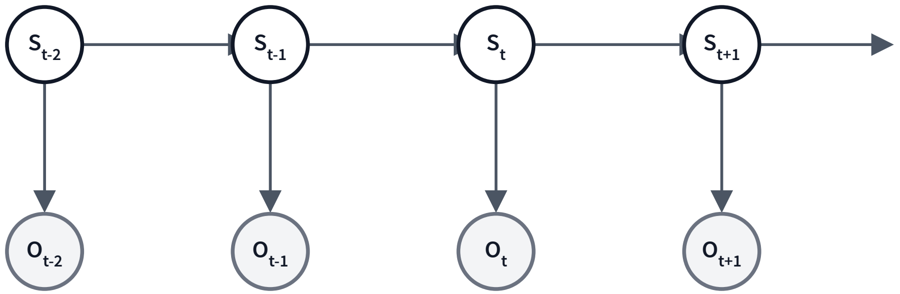
$P(S_K = T) = 0.5$ for all K – the chain has reached its steady state immediately.

Sensor (observation) model

Sensor Markov assumption

Each state is sufficient to generate its own observation:

$$P(O_t | S_t, S_{t-1}, \ldots, O_{t-1}, \ldots) = P(O_t | S_t).$$

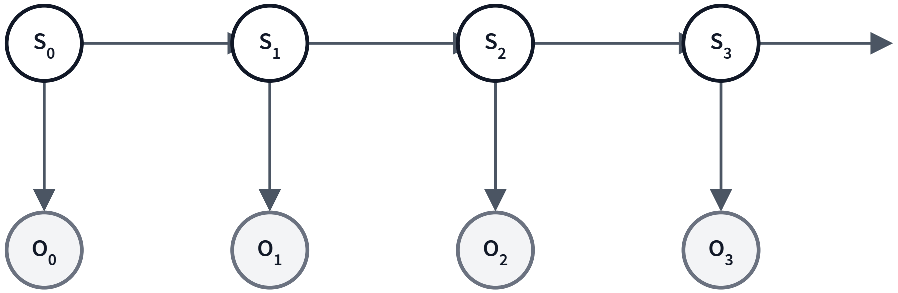


Sensor CPT

$$P(o_t | s_t) = 0.9$$
$$P(o_t | \neg s_t) = 0.2$$

Complete HMM for the umbrella story

$S_t = T$ iff it rains on day t. $O_t = T$ iff the director carries an umbrella on day t.



Prior & transition

$P(s_0) = 0.5$
$P(s_t|s_{t-1}) = 0.7$
$P(s_t|\neg s_{t-1}) = 0.3$

Sensor

$P(o_t|s_t) = 0.9$
$P(o_t|\neg s_t) = 0.2$

By the same reasoning as before, $P(O_K = T) = 0.5 \cdot 0.9 + 0.5 \cdot 0.2 = 0.55$ for all K (steady state, no evidence).

Hidden Markov Model

HMM

- A **Markov process** over time (state S_t depends only on S_{t-1}).
- States are **hidden**; we never observe them directly.
- Evidence O_t is **observable** and depends only on S_t.

An HMM is a Bayes net — so the **variable elimination algorithm** already works! But two specialized algorithms exploit the chain structure:

- **Forward-backward** — filtering & smoothing
- **Viterbi** — most-likely explanation

Four common inference tasks

☀ Filtering — *now*

What state am I in **right now**, given all evidence to date?

$$P(S_k | o_{0:k})$$

→ Prediction — *tomorrow*

What state will I be in in the **future**?

$$P(S_{k+j} | o_{0:k}), \quad j > 0$$

↺ Smoothing — *yesterday*

What state was I in at some **past** time?

$$P(S_j | o_{0:k}), \quad j < k$$

⚔ Most-likely explanation

Which **sequence** of states is most likely?

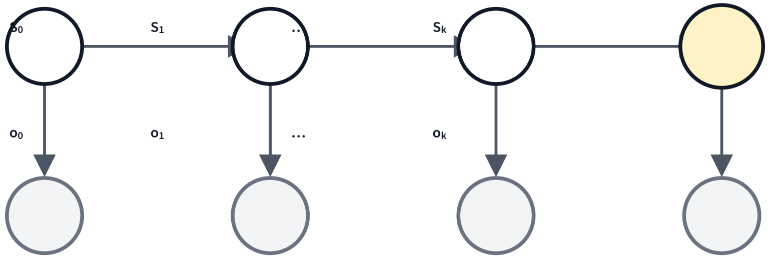
$$\arg\max_{s_{0:k}} P(s_{0:k} | o_{0:k})$$

Today: **filtering.**

Filtering

Given observations from time 0 to time k, what is my distribution over the state *now*?

$$P(S_k | o_{0:k})$$



Yellow = query. Gray = observed.

Naive: enumerate the joint

Tiny case ($k = 1$): $P(S_1 | o_0, \neg o_1) \propto$
$$\sum_{s_0} P(s_0)\, P(o_0 | s_0)\, P(S_1 | s_0)\, P(\neg o_1 | S_1)$$

3 mul × 2 + 1 add, all times 2 $\Rightarrow$ **14 ops** — manageable.

General case: marginalize *all* hidden states

$$P(S_k | o_{0:k}) \propto \sum_{s_0} \cdots \sum_{s_{k-1}} P(s_0)\, P(o_0 | s_0) \cdots P(o_k | S_k)$$

Naive enumeration

Sum over all 2^k hidden state assignments.

$O(k \cdot 2^k)$ **ops**

Forward recursion (today)

Reuse $f_{0:(k-1)}$ when computing $f_{0:k}$.

$O(k)$ **ops**

Filtering by forward recursion

Let $f_{0:k} \triangleq P(S_k | o_{0:k})$.

Base case (k = 0)

$$f_{0:0} = \alpha \, P(o_0 \mid S_0) \, P(S_0)$$

Recursive case (k ≥ 1)

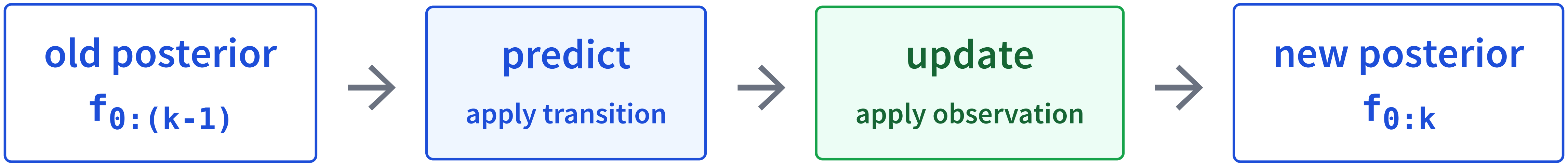
$$f_{0:k} = \alpha \, P(o_k \mid S_k) \sum_{s_{k-1}} P(S_k \mid s_{k-1}) \, f_{0:(k-1)}$$

Total cost: $O(k)$ ops — exponentially faster than naive enumeration.

α is a normalization constant (chosen so the two entries of $f_{0:k}$ sum to 1).

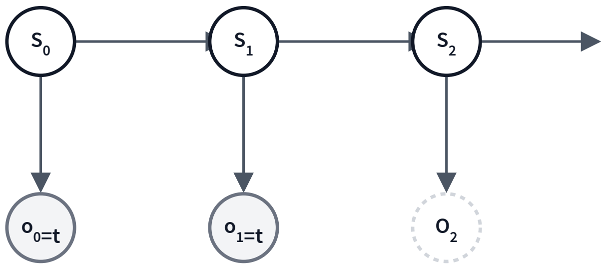
Filtering = predict + update

$$f_{0:k} = \underbrace{\alpha\, P(o_k|S_k)}_{\textbf{update}} \cdot \underbrace{\sum_{s_{k-1}} P(S_k|s_{k-1})\, f_{0:(k-1)}}_{\textbf{predict}}$$



Recursive Bayesian update — one step at a time.

A filtering example: setup



The umbrella story. Given $O_0 = t,\ O_1 = t$, compute $f_{0:0}$ and $f_{0:1}$ using forward recursion.

Numbers

$P(s_0) = 0.5$

$P(s_t|s_{t-1}) = 0.7,\ \ P(s_t|\neg s_{t-1}) = 0.3$

$P(o_t|s_t) = 0.9,\ \ P(o_t|\neg s_t) = 0.2$

Base case: $f_{0:0} = P(S_0|o_0)$

Use $f_{0:0} = \alpha\, P(o_0|S_0)\, P(S_0)$, treating each as a 2-vector $\langle T, F\rangle$.

$$f_{0:0} = \alpha\, \langle P(o_0|s_0),\; P(o_0|\neg s_0)\rangle \cdot \langle P(s_0),\; P(\neg s_0)\rangle$$

$$= \alpha\, \langle 0.9,\; 0.2\rangle \cdot \langle 0.5,\; 0.5\rangle$$

$$= \alpha\, \langle 0.45,\; 0.1\rangle$$

$$= \langle 0.818,\; 0.182\rangle$$

Normalize: $0.45/(0.45 + 0.1) \approx 0.818$; $0.1/0.55 \approx 0.182$.

Recursive case: $f_{0:1} = P(S_1 | o_0, o_1)$

$f_{0:1} = \alpha\, P(o_1|S_1) \sum_{s_0} P(S_1|s_0)\, f_{0:0}$ with $f_{0:0} = \langle 0.818, 0.182 \rangle$.

$$= \alpha\,\langle 0.9, 0.2\rangle \cdot \big(\langle 0.7, 0.3\rangle \cdot 0.818 + \langle 0.3, 0.7\rangle \cdot 0.182\big)$$

$$= \alpha\,\langle 0.9, 0.2\rangle \cdot \big(\langle 0.5726, 0.2454\rangle + \langle 0.0546, 0.1274\rangle\big)$$

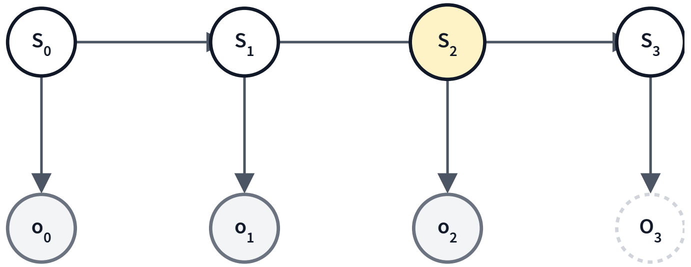
$$= \alpha\,\langle 0.9, 0.2\rangle \cdot \langle 0.6272, 0.3728\rangle$$

$$= \alpha\,\langle 0.56448, 0.07456\rangle$$

$$= \langle 0.883, 0.117\rangle$$

After two days of umbrellas, our belief that it rained today went from 0.5 $\longrightarrow$ 0.818 $\longrightarrow$ **0.883**.

Try it: 4-step HMM — compute $P(S_2 | o_0 \wedge o_1 \wedge o_2)$



Numbers

$P(s_0) = 0.4$

$P(s_t | s_{t-1}) = 0.7$

$P(s_t | \neg s_{t-1}) = 0.2$

$P(o_t | s_t) = 0.9$

$P(o_t | \neg s_t) = 0.2$

$$f_{0:0} = \langle 0.75, 0.25 \rangle \rightarrow f_{0:1} = \langle 0.859, 0.141 \rangle \rightarrow \boxed{f_{0:2} \approx \langle 0.884, 0.116 \rangle}$$

Deriving the filtering recursion

Six small steps from $P(S_k|o_{0:k})$ to the forward-recursion formula:

$$P(S_k|o_{0:k})$$

$$= P(S_k|o_k \land o_{0:(k-1)})$$ *rewrite*

$$= \alpha\, P(o_k|S_k \land o_{0:(k-1)})\, P(S_k|o_{0:(k-1)})$$ *Bayes' rule*

$$= \alpha\, P(o_k|S_k)\, P(S_k|o_{0:(k-1)})$$ *Markov (sensor)*

$$= \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k \land s_{k-1}|o_{0:(k-1)})$$ *sum rule*

$$= \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k|s_{k-1} \land o_{0:(k-1)})\, P(s_{k-1}|o_{0:(k-1)})$$ *chain rule*

$$= \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k|s_{k-1})\, f_{0:(k-1)}$$ *Markov (transition)*

Justify each step (1–3)

Pick one of: (A) rewrite (B) Bayes' rule (C) chain rule (D) Markov (E) sum rule

Q1.

$$P(S_k|o_{0:k}) = P(S_k|o_k \wedge o_{0:(k-1)})$$

A — rewrite

Q2.

$$= \alpha\, P(o_k|S_k \wedge o_{0:(k-1)})\, P(S_k|o_{0:(k-1)})$$

B — Bayes' rule

Q3.

$$= \alpha\, P(o_k|S_k)\, P(S_k|o_{0:(k-1)})$$

D — Markov

Justify each step (4–6)

Pick one of: (A) rewrite (B) Bayes' rule (C) chain rule (D) Markov (E) sum rule

Q4.

$$\ldots = \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k \wedge s_{k-1}|o_{0:(k-1)})$$

E — sum rule

Q5.

$$= \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k|s_{k-1} \wedge o_{0:(k-1)})\, P(s_{k-1}|o_{0:(k-1)})$$

C — chain rule

Q6.

$$= \alpha\, P(o_k|S_k) \sum_{s_{k-1}} P(S_k|$$

D — Markov

The big picture

$$f_{0:k} = \underbrace{\alpha\, P(o_k|S_k)}_{\textbf{update}} \cdot \underbrace{\sum_{s_{k-1}} P(S_k|s_{k-1})\, f_{0:(k-1)}}_{\textbf{predict}}$$

- **Predict:** roll forward the previous posterior through the transition model.
- **Update:** multiply by the new observation likelihood and renormalize.
- Each step is $O(1)$ in k (constant-size factors). Total: $O(k)$.

Learning goals (recap)

- ✓ **Construct** a hidden Markov model from a real-world scenario.
- ✓ State the **independence assumptions** baked into an HMM.
- ✓ Compute the **filtering probability** for a time step.
- ✓ **Justify** each step in the derivation of the filtering formula.

Next: HMMs, Part 2

- **Prediction** — what state will I be in tomorrow?
- **Smoothing** — what was my state yesterday, knowing today's evidence?
- **Most-likely explanation** — the *Viterbi* algorithm.

All three are variations on the same predict-update recursion.