Reinforcement Learning

Prof. Christian Bauckhage

lecture 06

Markov Chains and Hidden Markov Models

previously, we saw that PFSMs are *Markov chains* in disguise

in this lecture, we revisit this point and study the fundamentals of *Markov chains*

we then generalize them towards *hidden Markov models (HMMs)* and briefly look at what these are all about

in doing so, we will have our first contact with the unfathomable important idea of *dynamic programming*

outline

recap

hierarchy of behaviors

strategic
long term goals

tactical
mid term goals

reactive
short term goals

resources

contextual / world knowledge

information

reasoning / planning

computation time

deterministic finite state machine / automaton

$\Leftrightarrow$ a tuple

$$\boldsymbol{M} = \big(\mathcal{S}, \mathcal{A}, \delta, S_0, \mathcal{F}\big)$$

where

$$\mathcal{S} \equiv \text{finite set of states}$$

$$\mathcal{A} \equiv \text{finite set of actions}$$

$$\delta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \equiv \text{state transition function}$$

$$S_0 \in \mathcal{S} \equiv \text{initial state}$$

$$\mathcal{F} \subseteq \mathcal{S} \equiv \text{set of final states}$$

graphical models / Bayesian networks $\Leftrightarrow$ *"conditional independence diagrams"*

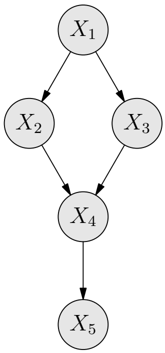
joint probability

$$\begin{aligned} &p(X_1, X_2, \ldots, X_5) \\ &= p(X_1) \cdot p(X_2 \mid X_1) \cdot p(X_3 \mid X_1) \\ &\qquad \cdot p(X_4 \mid X_2, X_3) \cdot p(X_5 \mid X_4) \end{aligned}$$

a joint probability *respects* an ordered DAG, if

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid \mathcal{V}_{Pred(i)})$$

ordered DAG



Markov models / Markov chains

question

what are Markov models ?

answer

simple representations of dependencies in
temporal or sequential data or processes

question

what are Markov models ?

answer

simple representations of dependencies in temporal or sequential data or processes

the incarnation of the reasonable idea that

the future is independent of the past given the present

question

why study Markov models ?

answer

in general, Markov models allow for

representing temporal patterns that are subject to chance

probabilistic reasoning about sequences of observations

w.r.t. artificial agents, Markov models provide

“simple” solutions for behavior modeling and recognition

general idea

consider a set of rvs $\{S_t\}_{t \in \mathbb{N}}$ and assume that t indicates a point in time

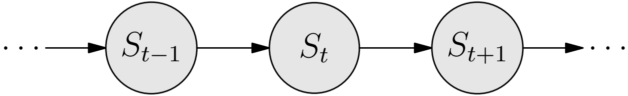
general idea

consider a set of rvs $\{S_t\}_{t \in \mathbb{N}}$ and assume that t indicates a point in time

now, if *the future is independent of the past given the present*, that is if

$$\left(S_{t+1} \perp\!\!\!\perp S_{t-1}\right) \mid S_t$$

then $p\left(S_0, S_1, S_2, \ldots\right)$ must respect



questions

temporal processes ?

why several S_t ?

why not just S ?

answers

say we want to reason about the weather

we may consider a single rv $S \Leftrightarrow$ *the weather*

if $S \in \{sunny, cloudy, rainy, snowy\}$, we can evaluate $p(S = cloudy)$

but how to express belief in that it will snow tomorrow given that it rains today ?

answers

say we want to reason about the weather

we may consider a single rv $S \Leftrightarrow$ *the weather*

if $S \in \{sunny, cloudy, rainy, snowy\}$, we can evaluate $p(S = cloudy)$

but how to express belief in that it will snow tomorrow given that it rains today ?

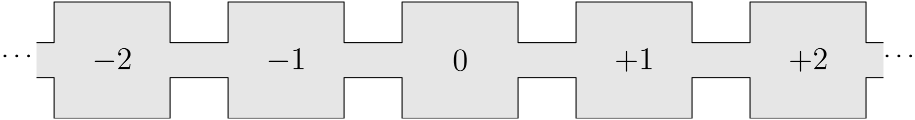
if we instead consider a set of rvs $\{S_t\}_{t \in \mathbb{N}}$ where

$$
\begin{aligned}
S_t &\Leftrightarrow \textit{the weather today} \\
S_{t+1} &\Leftrightarrow \textit{the weather tomorrow}
\end{aligned}
$$

we can immediately reason about $p(S_{t+1} = snowy \mid S_t = rainy)$

motivating example

imagine an *infinite* sequence of rooms and an agent living in this environment



consider a set of rvs $\{S_t\}_{t \in \mathbb{N}}$ where $S_t \in \mathbb{Z}$ and let

$$S_t = s \quad \Leftrightarrow \quad \text{at time } t\text{, the agent is in room } s$$

at $t = 0$, the agent starts in $s = 0$ and then executes either

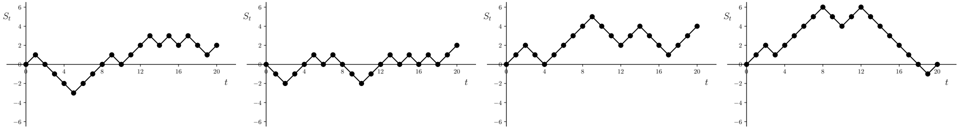
first behavior

$$p\big(S_t = s+1 \mid S_{t-1} = s\big) = 1/2$$
$$p\big(S_t = s-1 \mid S_{t-1} = s\big) = 1/2$$

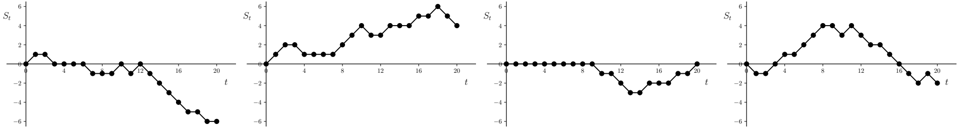
second behavior

$$p\big(S_t = s+1 \mid S_{t-1} = s\big) = 1/3$$
$$p\big(S_t = s \mid S_{t-1} = s\big) = 1/3$$
$$p\big(S_t = s-1 \mid S_{t-1} = s\big) = 1/3$$

instances of the first behavior



instances of the second behavior



do you see a characteristic difference ?

remarks

both behaviors are random walks over the integers $\mathbb{Z}$

random walks are examples of stochastic processes

$\Leftrightarrow$ a system evolves / an agent behaves *randomly* but *not arbitrarily*

random walks can be generalized to $\mathbb{R}^d$ and to infinitesimal steps $t \to 0$

examples of generalizations are *Brownian motion* or *Wiener processes*

(the global finance industry heavily relies on working with such models)

w.r.t. the above example, we observe

- by definition, S_t depends on S_{t-1}
- but, to some extent, also on $S_{t-2}, S_{t-3}, \ldots, S_0$
- why?

w.r.t. the above example, we observe

- by definition, S_t depends on S_{t-1}
- but, to some extent, also on $S_{t-2}, S_{t-3}, \ldots, S_0$
- why? because, for instance, S_3 cannot be 5000 if $S_0 = 0$

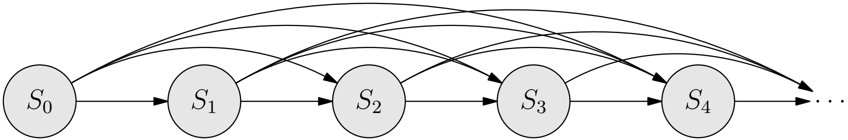
w.r.t. the above example, we observe

by definition, S_t depends on S_{t-1}

but, to some extent, also on $S_{t-2}, S_{t-3}, \ldots, S_0$

why? because, for instance, S_3 cannot be 5000 if $S_0 = 0$

$\Rightarrow$ the most general graphical model would thus look like this



note

this will get out of hand $\Leftrightarrow$ is too general

note

this will get out of hand $\Leftrightarrow$ is too general

$\Rightarrow$ a “reasonable” simplifying assumption is

the recent past
is *more informative*
than the distant past

take home messages

a “reasonable” modeling assumption is that

S_t only depends on $S_{t-1}, S_{t-2}, \ldots, S_{t-\tau}$

in this case, the conditional probability becomes

$$p\big(S_t \bigm| S_{t-1}, \ldots, S_{t-\tau}, \ldots, S_0\big) = p\big(S_t \bigm| S_{t-1}, \ldots, S_{t-\tau}\big)$$

take home messages

a “reasonable” modeling assumption is that

S_t only depends on $S_{t-1}, S_{t-2}, \ldots, S_{t-\tau}$

in this case, the conditional probability becomes

$$p\big(S_t \bigm| S_{t-1}, \ldots, S_{t-\tau}, \ldots, S_0\big) = p\big(S_t \bigm| S_{t-1}, \ldots, S_{t-\tau}\big)$$

the simplest possible model would set $\tau = 1$, such that

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

discrete state space

$\Leftrightarrow$ a finite / countably infinite set $\mathcal{S} = \{s_1, s_2, \ldots\}$

stochastic process

$\Leftrightarrow$ a set of rvs $\{S_t \mid t \in \mathbb{N}\}$ where each $S_t \in \mathcal{S}$

discrete time Markov chain (DTMC)

$\Leftrightarrow$ a stochastic process whose joint probability respects

$$\cdots \longrightarrow S_{t-1} \longrightarrow S_t \longrightarrow S_{t+1} \longrightarrow \cdots$$

$\Leftrightarrow$ a stochastic process such that for all time points $t \in \mathbb{N}$

$$p\left(S_t \mid S_{t-1}, \ldots, S_0\right) = p\left(S_t \mid S_{t-1}\right)$$

time homogeneous DTMC

$\Leftrightarrow$ a DTMC where

$$p\big(S_t = s_j \bigm| S_{t-1} = s_i\big) = p\big(S_1 = s_j \bigm| S_0 = s_i\big) \quad \forall\, t \in \mathbb{N}$$

note

$\Rightarrow$ **in a time homogeneous MC, the probability for the process to transit from state s_i to state s_j does *not* depend on time t**

we therefore write

$$p\big(S_t = s_j \mid S_{t-1} = s_i\big) = p\big(s_j \mid s_i\big) = p_{ji}$$

note

we write

$$p_{ji} = p\left(s_j \mid s_i\right) \tag{1}$$

to express the transition probability $s_i \rightarrow s_j$

most of *the literature* does it the other way around and writes

$$p_{ij} = p\left(s_j \mid s_i\right) \tag{2}$$

to express the transition probability $s_i \rightarrow s_j$

note

we write

$$p_{ji} = p\left(s_j \mid s_i\right) \tag{1}$$

to express the transition probability $s_i \to s_j$

most of *the literature* does it the other way around and writes

$$p_{ij} = p\left(s_j \mid s_i\right) \tag{2}$$

to express the transition probability $s_i \to s_j$

(1) and (2) are subtly different . . . there are no typos here !!!

note

if we are dealing with a finite set $\mathcal{S}$ where

$$|\,\mathcal{S}\,| = n$$

we can arrange the p_{ji} in a matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

note

we have $p_{ji} \geqslant 0$ and

$$\sum_j p_{ji} = 1$$

the latter is because

$$\sum_j p_{ji} = \sum_j p\big(s_j \mid s_i\big) = \sum_j \frac{p\big(s_j, s_i\big)}{p\big(s_i\big)} = \frac{p\big(s_i\big)}{p\big(s_i\big)} = 1$$

$\Rightarrow$ *our* matrix $\boldsymbol{P}$ is *column stochastic*

(in the literature, $\boldsymbol{P}$ is often *row stochastic*)

there are 2 graphical representations
of Markov chains

conditional independence diagrams

this is the point of view we have considered so far

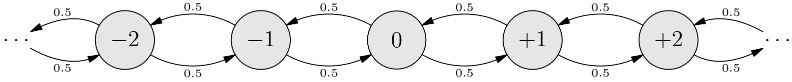
for example, for our first random walk on $\mathbb{Z}$, we have

$$S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow S_3 \longrightarrow \cdots$$

state transition diagrams

if $p\big(S_t = s_j \bigm| S_{t-1} = s_i\big) = p\big(s_j \bigm| s_i\big) = p_{ji}$, the state transition probabilities p_{ji} fully characterize the behavior of the chain

for example, for our first random walk on $\mathbb{Z}$, we have



note

state transition diagrams are eerily reminiscent of *finite state machines*

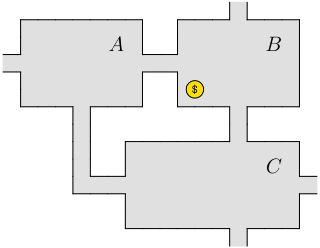
indeed, time homogeneous DTMCs are a kind of non-deterministic state machines

they model / represent processes where state transitions happen in a random fashion

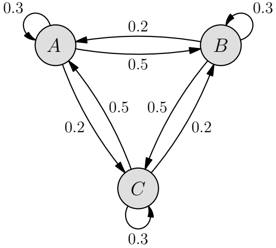
to stress it again: **state transitions are subject to uncertainty but not arbitrary**

game AI example

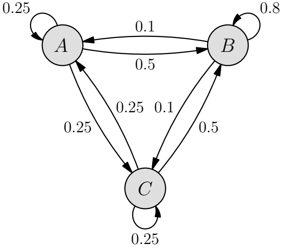
consider a state space $\mathcal{S} = \{A, B, C\}$ and two Markov chains λ_1 and λ_2 over $\mathcal{S}$



an environment



Markov chainλ_1



Markov chain λ_2

exemplary behaviors

two samples from- or realizations of Markov chain λ_1

$CCABABABCCCACABBAAABBBBABCBCABACBBCCCABCABCABCAAAB \cdots$

$BCBBABABBBBBCACCABBBCBABCBCAACBABCBBCAABCBCCACBBAA \cdots$

two samples from- or realizations of Markov chain λ_2

$BBBCBBBBBBBBAAAABBBCABBBBBBAACACABBBBBBBBBBBBBACCB \cdots$

$BBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBCBBBCACBBBABBABB \cdots$

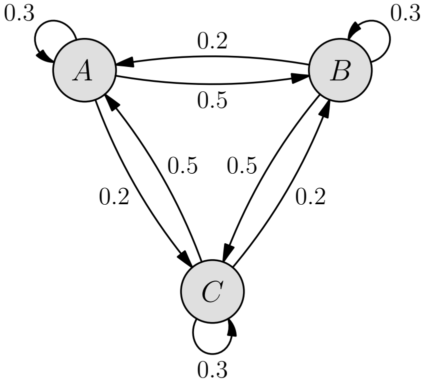
$\Rightarrow$ once again: **stochastic process** $\Leftrightarrow$ **structured randomness**

question

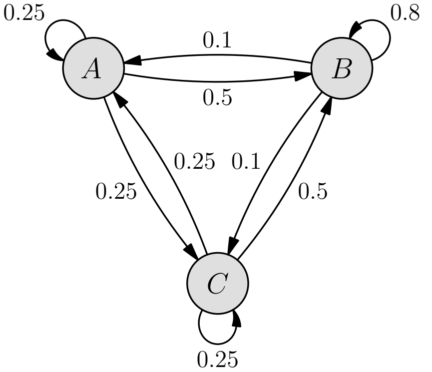
what kind of behaviors do the two chains represent ?

answer

we arguably have . . .



$\lambda_1 \Leftrightarrow$ patrolling



$\lambda_2 \Leftrightarrow$ guarding

now, assume we are given a sequence of observations of how an agent has been moving through the environment

$$os = A : A : B : B : B : B : C : B : B : A \equiv AABBBBCBBA$$

can we "recognize" what this agent has been up to?

now, assume we are given a sequence of observations of how an agent has been moving through the environment

$$os = A : A : B : B : B : B : C : B : B : A \equiv AABBBBCBBA$$

can we “recognize” what this agent has been up to?

yes, we can! we simply have to compute the values

$$p\big(os \mid \lambda_1\big) \qquad \text{and} \qquad p\big(os \mid \lambda_2\big)$$

for λ_1, we have

$$
\begin{aligned}
p\big(AABBBBCBBA\big) &= p\big(S_0 = A\big) \cdot p\big(S_1 = A \mid S_0 = A\big) \\
&\quad \cdot p\big(S_2 = B \mid S_1 = A\big) \cdots p\big(S_9 = A \mid S_8 = B\big) \\
&= p\big(S_0 = A\big) \cdot p_{AA} \cdot p_{BA} \cdots p_{AB} \\
&= {\color{red}1} \cdot 0.30 \cdot 0.5 \cdot 0.3 \cdot 0.3 \cdot 0.3 \cdot 0.5 \cdot 0.2 \cdot 0.3 \cdot 0.2 \approx 0.000025
\end{aligned}
$$

for λ_2, we have

$$
p\big(AABBBBCBBA\big) = {\color{red}1} \cdot 0.25 \cdot 0.5 \cdot 0.8 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.5 \cdot 0.8 \cdot 0.1 \approx 0.00026
$$

for λ_1, we have

$$
\begin{aligned}
p(AABBBBCBBA) &= p(S_0 = A) \cdot p(S_1 = A \mid S_0 = A) \\
&\quad \cdot p(S_2 = B \mid S_1 = A) \cdots p(S_9 = A \mid S_8 = B) \\
&= p(S_0 = A) \cdot p_{AA} \cdot p_{BA} \cdots p_{AB} \\
&= {\color{red}1} \cdot 0.30 \cdot 0.5 \cdot 0.3 \cdot 0.3 \cdot 0.3 \cdot 0.5 \cdot 0.2 \cdot 0.3 \cdot 0.2 \approx 0.000025
\end{aligned}
$$

for λ_2, we have

$$
p(AABBBBCBBA) = {\color{red}1} \cdot 0.25 \cdot 0.5 \cdot 0.8 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.5 \cdot 0.8 \cdot 0.1 \approx 0.00026
$$

$\Rightarrow$ w.r.t. the two models, it appears that the agent is guarding the treasure in room B

we can also use Markov chains to anticipate the future

for instance, now that we “know” that the agent guards the treasure, we might ask:

> if the agent is currently in room B, how likely will it stay there for the next τ time steps ?

we can also use Markov chains to anticipate the future

for instance, now that we "know" that the agent guards the treasure, we might ask:

> if the agent is currently in room B, how likely will it stay there for the next τ time steps ?

to answer this particular question, we compute the value

$$
\begin{aligned}
& p\big(S_t = B,\, S_{t+1} = B,\, \ldots, S_{t+\tau} = B,\, S_{t+\tau+1} \neq B\big) \\
&= p\big(S_{t+1} = B \bigm| S_t = B\big) \cdots p\big(S_{t+\tau+1} \neq B \bigm| S_{t+\tau} = B\big) \\
&= p_{BB}^{\tau} \cdot (1 - p_{BB})
\end{aligned}
$$

hidden Markov models (HMMs)

note

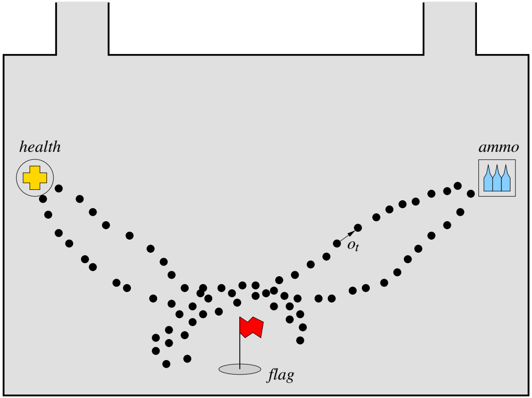
sometimes (often), we cannot observe the

$$S_t = s \in \mathcal{S} = \{s_i\}_{i=1}^{n}$$

directly but only have indirect observations

$$O_t = \boldsymbol{o} \in \mathcal{O} \subset \mathbb{R}^d$$

game AI example



$$\mathcal{S} = \{\text{guard flag, run for health, run for ammo, return to flag}\} = \{s_1, s_2, s_3, s_4\}$$

hidden Markov models can cope with situations like this hidden Markov model (HMM)

$\Leftrightarrow$ a graphical model with two sets of rvs

states

$$S_1, S_2, \ldots, S_T \in \mathcal{S}$$

data / observations

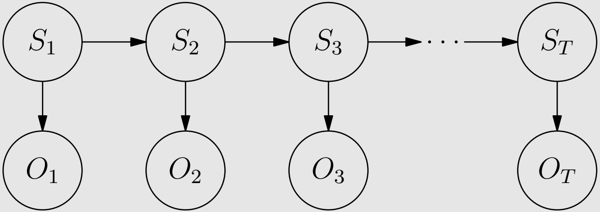
$$O_1, O_2, \ldots, O_T \in \mathcal{O}$$

S_t are *hidden* or *latent variables*, O_t are *observable variables*

for HMMs, it is common to count time t from $1 \ldots$

hidden Markov model (HMM)

⇔ a graphical model where



$$p(S_1, \ldots, S_n, O_1, \ldots, O_n) = p(S_1)\, p(O_1 \mid S_1) \prod_{t=2}^{T} p(S_t \mid S_{t-1})\, p(O_t \mid S_t)$$

components of a HMM

initial state distribution

$$p\big(S_1 = s_j\big) = p\big(s_j\big) = \pi_j \qquad \Leftrightarrow \qquad \text{stochastic vector } \boldsymbol{\pi} \in \mathbb{R}^n$$

transition probabilities

$$p\big(S_t = s_j \;\big|\; S_{t-1} = s_i\big) = a_{ji} \qquad \Leftrightarrow \qquad \text{stochastic matrix } \boldsymbol{A} \in \mathbb{R}^{n \times n}$$

emission probabilities

$$p\big(O_t = \boldsymbol{o} \;\big|\; S_t = s_j\big) = b_j(\boldsymbol{o}) \qquad \Leftrightarrow \qquad \text{set of pdfs } \mathcal{B} = \big\{b_j(\boldsymbol{o})\big\}_{j=1}^{n}$$

hidden Markov model

$\Leftrightarrow$ a triple

$$\lambda = \left(\boldsymbol{\pi}, \boldsymbol{A}, \mathcal{B}\right)$$

such that

$$p\left(s_{i_1}, \ldots, s_{i_T}, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\right) = \pi_{i_1} \, b_{i_1}\left(\boldsymbol{o}_1\right) \prod_{t=2}^{T} a_{i_t, i_{t-1}} \, b_{i_t}\left(\boldsymbol{o}_t\right)$$

note

there are 3 fundamental questions when working with HMMs . . .

1) given many *training sequences* $\boldsymbol{O} = \boldsymbol{o}^1_{1:T_1}, \boldsymbol{o}^2_{1:T_2}, \boldsymbol{o}^3_{1:T_3}, \ldots$, how to find parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \mathcal{B})$ of an HMM λ so as to maximize $p(\boldsymbol{O} \mid \lambda)$?

$\Leftrightarrow$ **how to train a hidden Markov model from data ?**

2) given a sequence of observations $\boldsymbol{o}_{1:T} = \boldsymbol{o}_1 : \boldsymbol{o}_2 : \cdots : \boldsymbol{o}_T$ and a trained HMM $\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \mathcal{B})$, how to compute $p(s_{i_1}, \ldots, s_{i_T} \mid \boldsymbol{o}_{1:T}, \lambda)$?

$\Leftrightarrow$ **how to explain data in terms of latent states ?**

3) given a sequence of observations $\boldsymbol{o}_{1:T} = \boldsymbol{o}_1 : \boldsymbol{o}_2 : \cdots : \boldsymbol{o}_T$ and a trained HMM $\lambda = (\boldsymbol{\pi}, \boldsymbol{A}, \mathcal{B})$, how to compute $p(\boldsymbol{o}_{1:T} \mid \lambda)$?

$\Leftrightarrow$ **how well does λ explain data ?**

question

how are HMMs used in practice ?

answer

...

goal: be able to recognize $1 \leqslant l \leqslant L$ different temporal patterns

for training: collect (a lot of) training examples for each pattern (e.g. movement data for different behaviors from many players)

in training: train a HMM for each pattern

fix the number m of hidden states, and run the Baum-Whelch algorithm to learn $\lambda_l = (\boldsymbol{\pi}_l, \boldsymbol{A}_l, \mathcal{B}_l)$

for each λ_l, apply the Viterbi algorithm to evaluate what kind of states were determined and refine λ_l if necessary

in application: for a given sequence $\boldsymbol{o}_{1:T}$, determine

$$\lambda_* = \underset{\lambda_l}{\operatorname{argmax}}\; p\big(\boldsymbol{o}_{1:T} \,\big|\, \lambda_l\big)$$

disclaimer

in this course, we may ignore HMM training and inspection

in what follows, we therefore focus on model application . . .

the forward algorithm

HMMs for temporal pattern recognition

given different HMMs $\lambda_1, \ldots, \lambda_L$ which model different classes of temporal patterns together with an observation

$$\boldsymbol{o}_{1:T} = \boldsymbol{o}_1 : \boldsymbol{o}_2 : \cdots : \boldsymbol{o}_T$$

determine the model λ_* which best explains the observation

$$\lambda_* = \underset{\lambda_l}{\mathrm{argmax}} \; p\big(\boldsymbol{o}_{1:T} \;\big|\; \lambda_l\big)$$

finally, classify $\boldsymbol{o}_{1:T}$ to belong to the class represented by λ_*

questions

how to compute λ_* ?

$\Leftrightarrow$ how to compute $p\big(\boldsymbol{o}_{1:T} \;\big|\; \lambda\big)$?

$\Leftrightarrow$ how to compute $p\big(\boldsymbol{o}_{1:T}\big)$ for a given λ ?

naïve approach (marginalization over all states)

$$
\begin{aligned}
p\big(\boldsymbol{o}_{1:T}\big) &= \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \dots \sum_{i_T=1}^{n} p\big(\boldsymbol{o}_{1:T}, s_{i_1}, s_{i_2}, \dots, s_{i_T}\big) \\
&\overset{!}{=} \sum_{s_{i_1:i_T}} p\big(\boldsymbol{o}_{1:T}, s_{i_1:i_T}\big) \\
&= \sum_{s_{i_1:i_T}} p\big(\boldsymbol{o}_{1:T} \mid s_{i_1:i_T}\big)\, p\big(s_{i_1:i_T}\big)
\end{aligned} \tag{3}
$$

(3) is great, because all its ingredients are known

$$p\left(s_{i_1:i_T}\right) = p\left(s_{i_1}\right) \prod_{t=2}^{T} p\left(s_{i_t} \mid s_{i_{t-1}}\right)$$

$$p\left(\boldsymbol{o}_{1:T} \mid s_{i_1:i_T}\right) = \frac{p\left(\boldsymbol{o}_{1:T}, s_{i_1:i_T}\right)}{p\left(s_{i_1:i_T}\right)} = \frac{p\left(\boldsymbol{o}_1 \mid s_{i_1}\right) p\left(s_{i_1}\right) \prod_{t=2}^{T} p\left(s_{i_t} \mid s_{i_{t-1}}\right) p\left(\boldsymbol{o}_t \mid s_{i_t}\right)}{p\left(s_{i_1}\right) \prod_{t=2}^{T} p\left(s_{i_t} \mid s_{i_t-1}\right)}$$

$$= \prod_{t=1}^{T} p\left(\boldsymbol{o}_t \mid s_{i_t}\right)$$

(3) is bad, because it sums over all possible sequences $s_{i_1:i_T}$ of length T

this is really bad, because, for $|\mathcal{S}| = n$, there exist n^T such sequences

$\Rightarrow$ in most scenarios, evaluating (3) is infeasible even on supercomputers

so, let's switch gears . . .

note

we may consider

$$p(\boldsymbol{o}_{1:T}) = \sum_{i_T=1}^{n} p(s_{i_T}, \boldsymbol{o}_{1:T})$$

(this drops out of thin air but is fine !!!)

$$\equiv \sum_{i_T=1}^{n} \alpha_T(i_T)$$

note

we may consider

$$p(\boldsymbol{o}_{1:T}) = \sum_{i_T=1}^{n} p(s_{i_T}, \boldsymbol{o}_{1:T})$$

(this drops out of thin air but is fine !!!)

$$\equiv \sum_{i_T=1}^{n} \alpha_T(i_T)$$

alas, this expression does not contain any of the given probabilities / distributions . . .

clever trick

$$\alpha_T(i_T) = p(s_{i_T}, \boldsymbol{o}_{1:T}) = \sum_{i_{T-1}} p(s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T})$$

clever trick

$$\alpha_T(i_T) = p\left(s_{i_T}, \boldsymbol{o}_{1:T}\right) = \sum_{i_{T-1}} p\left(s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T}\right)$$

$$= \sum_{i_{T-1}} p\left(\boldsymbol{o}_T \mid s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T-1}\right) p\left(s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T-1}\right)$$

clever trick

$$\alpha_T(i_T) = p(s_{i_T}, \boldsymbol{o}_{1:T}) = \sum_{i_{T-1}} p(s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T})$$

$$= \sum_{i_{T-1}} p(\boldsymbol{o}_T \mid s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})\, p(s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})$$

$$= \sum_{i_{T-1}} \underbrace{p(\boldsymbol{o}_T \mid s_{i_T}, s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})}_{(\boldsymbol{o}_T \perp\!\!\!\perp s_{i_{T-1}}, \boldsymbol{o}_{1:T-1}) \mid s_{i_T}}\, \underbrace{p(s_{i_T} \mid s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})}_{(s_{i_T} \perp\!\!\!\perp \boldsymbol{o}_{1:T-1}) \mid s_{i_{T-1}}}\, p(s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})$$

$$= \sum_{i_{T-1}} \underbrace{p(\boldsymbol{o}_T \mid s_{i_T})}_{known}\, \underbrace{p(s_{i_T} \mid s_{i_{T-1}})}_{known}\, \underbrace{p(s_{i_{T-1}}, \boldsymbol{o}_{1:T-1})}_{\alpha_{T-1}(i_{T-1})}$$

consequence

$$p(\boldsymbol{o}_{1:T}) = \sum_{j=1}^{n} \alpha_T(j) = \sum_{j=1}^{n} \sum_{i=1}^{n} p(\boldsymbol{o}_T \mid s_j)\, p(s_j \mid s_i)\, \alpha_{T-1}(i)$$

$$p(\boldsymbol{o}_1) = \sum_{j=1}^{n} \alpha_1(j) = \sum_{j=1}^{n} p(s_j, \boldsymbol{o}_1) = \sum_{j=1}^{n} p(\boldsymbol{o}_1 \mid s_j)\, p(s_j)$$

forward algorithm

function *Forward*$(\boldsymbol{o}_{1:T}, (\boldsymbol{\pi}, \boldsymbol{A}, \mathcal{B}))$

allocate matrix $\boldsymbol{\alpha} \in \mathbb{R}^{n \times T}$

for $j = 1, \ldots, n$

$$\alpha_{j,1} = b_j(\boldsymbol{o}_1)\, \pi_j$$

for $t = 2, \ldots, T$
  for $j = 1, \ldots, n$

$$\alpha_{j,t} = b_j(\boldsymbol{o}_t) \sum_{i=1}^{n} a_{ji}\, \alpha_{i,t-1}$$

return $\sum_{j=1}^{n} \alpha_{j,T}$

note

$\Rightarrow$ running the *forward algorithm*, estimation efforts reduce from $O\left(n^T\right)$ to $O\left(T \cdot n^2\right)$

the forward algorithm is an example for the power of *dynamic programming*

dynamic programming methods were introduced by **Richard Bellman** in the 1950s

the fundamental idea is to recursively rewrite complex problems in terms of simpler ones, i.e. to reuse information computed earlier at later stages of a computation

note: this slide is extremely foreshadowing !

summary

we now know about

discrete space and time Markov chains

state transition probabilities and -diagrams

the use of Markov chains for classification or
prediction of temporal patterns (i.e. behaviors)

hidden Markov models (HMMs)

the forward algorithm for temporal pattern recognition

the benefits of conditional independence in (complex) probabilistic models

the use of dynamic programming for solving seemingly impossible problems