Probabilistic Graphical Models

Saeed Saremi

Assigned reading: Chapter 11,
[Barber12] (23.2.2 & 23.2.5)[1]

November 5, 2024

[1]Barber, D. *Bayesian Reasoning and Machine Learning.* Cambridge University Press, 2012.
`http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/090310.pdf`

OUTLINE

- Probabilistic graphical models
  - finish the D-SEPARATION examples
  - Conditional probability table
  - Inference in PGMs
- Inference in HMMs
  - DYNAMIC PROGRAMMING (a review)
  - α-update algorithm
  - VITERBI algorithm

- To check conditional independence between X_i and X_j, conditioned on a set of nodes $\mathcal{C}$, consider all *indirected* paths between X_i and X_j.
- We can declare $X_i \perp\!\!\!\perp X_j \mid \mathcal{C}$ if all paths are blocked.
- A path is blocked if *any node* in the path is blocked (via "atomic" triples).

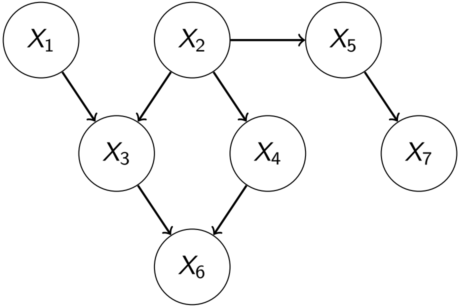


Figure: $X_1 \perp\!\!\!\perp X_2 \mid X_6$?

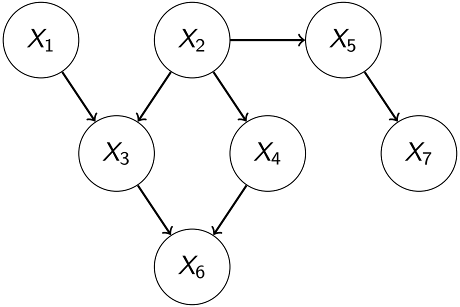
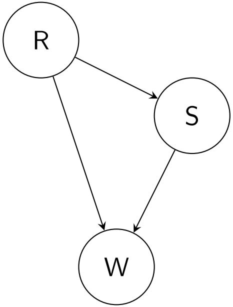


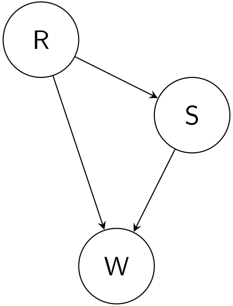
Figure: $X_3 \perp\!\!\!\perp X_7 \mid X_2$?

JOINT PROBABILITY DISTRIBUTION: AN EXAMPLE



S	R	W	$P(S, R, W)$
T	T	T	
T	T	F	
T	F	T	
T	F	F	
F	T	T	
F	T	F	
F	F	T	
F	F	F	

- Canonical example is a multivariate Gaussian. The joint probability is specified by the mean, a d-dimensional vector, and the covariance matrix, a $d \times d$ symmetric matrix.
- Suppose we have d binary random variables. Then the joint distribution can be specified by a table with 2^d entries. This quickly becomes intractable, both for specification, and subsequently in estimation from data.
- The secret to tractability is conditional independence. This information can be captured by a directed acyclic graph (DAG). For such a graph, every node has well-defined parents and the joint distribution is the product of "local" conditional distributions.



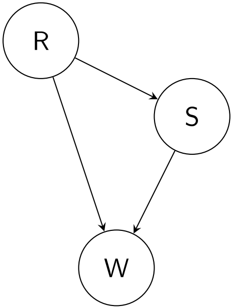
$$P(R,S,W) = P(R)P(S|R)P(W|S,R)$$

$P(R)$	$R = \text{True}$	$R = \text{False}$
	0.2	0.8

$P(S\|R)$	$S = \text{True}$	$S = \text{False}$
$R = \text{True}$	0.01	0.99
$R = \text{False}$	0.4	0.6

$P(W\|S,R)$	$W = \text{True}$	$W = \text{False}$
$S = \text{True}, R = \text{True}$	0.99	0.01
$S = \text{True}, R = \text{False}$	0.8	0.2
$S = \text{False}, R = \text{True}$	0.9	0.1
$S = \text{False}, R = \text{False}$	0.0	1.0

$P(R)$	$R = \text{True}$	$R = \text{False}$
	0.2	0.8



$P(S\|R)$	$S = \text{True}$	$S = \text{False}$
$R = \text{True}$	0.01	0.99
$R = \text{False}$	0.4	0.6

$P(W\|S,R)$	$W = \text{True}$	$W = \text{False}$
$S = \text{True}, R = \text{True}$	0.99	0.01
$S = \text{True}, R = \text{False}$	0.8	0.2
$S = \text{False}, R = \text{True}$	0.9	0.1
$S = \text{False}, R = \text{False}$	0.0	1.0

S	R	W	$P(S,R,W)$
T	T	T	
T	T	F	
T	F	T	
T	F	F	
F	T	T	
F	T	F	
F	F	T	
F	F	F	

PGM = $\prod$ CPT

$$P(X_1, \ldots, X_d) = \prod_{i=1}^{d} P(X_i \mid \mathrm{pa}(X_i))$$

In general

$$|\mathrm{pa}(X_i)| \ll d,$$

therefore this leads to a much more compact representation of the joint probability.

THE JOINT PROBABILITY DISTRIBUTION = THE WORLD

$$P(R=T \mid W=T) = \frac{P(R=T, W=T)}{P(W=T)} = \frac{\sum_{s\in\{T,F\}} P(R=T, S=s, W=T)}{\sum_{r,s\in\{T,F\}} P(R=r, S=s, W=T)}$$

$$P(R = T \mid W = T) = \frac{P(R = T, W = T)}{P(W = T)} = \frac{\sum_{s \in \{T,F\}} P(R = T, S = s, W = T)}{\sum_{r,s \in \{T,F\}} P(R = r, S = s, W = T)}$$

We can calculate any term in the numerator and denominator using our factorization, e.g.,

$$\begin{aligned} P(R = T, S = T, W = T) &= P(R = T)P(S = T \mid R = T)P(W = T \mid S = T, R = T) \\ &= 0.2 \times 0.01 \times 0.99 \\ &= 0.00198 \end{aligned}$$

Then the numerical results (check at home!) are:

$$P(R = T \mid W = T) = \frac{0.00198 + 0.1584}{0.00198 + 0.288 + 0.1584 + 0} \approx 0.36$$

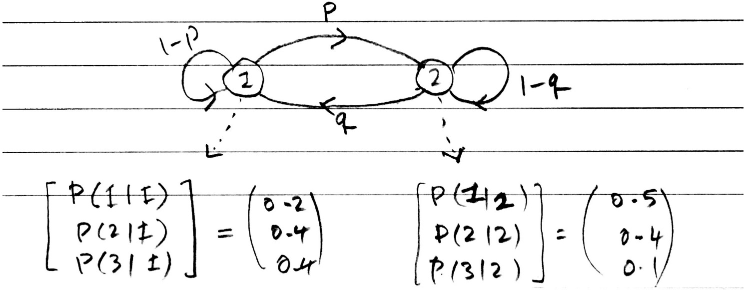
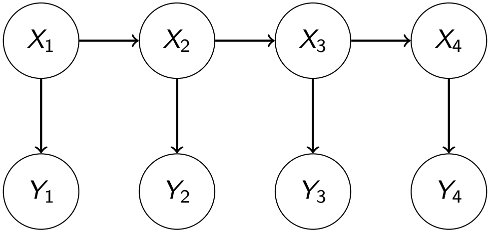


Figure: "stochastic automaton" representation of the HMM



$$P(X_{1:T}, Y_{1:T}) = \prod_{t=1}^{T} P(X_t | X_{t-1}) P(Y_t | X_t)$$

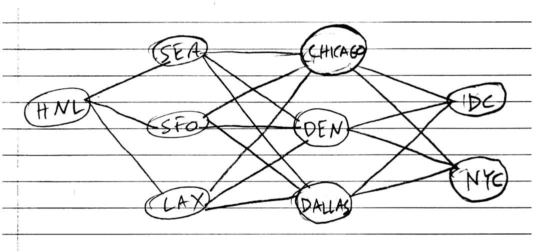
We are interested in various inference problems, e.g.:

- How can we reason about $P(X_T | Y_{1:T})$?
- What are the most likely states given the observations $Y_{1:T}$, i.e., find

$$\underset{X_{1:T}}{\text{argmax}}\, P(X_{1:T} \mid Y_{1:T}).$$

DYNAMIC PROGRAMMING

EXAMPLE: FIND CHEAPEST FLIGHT



- K choices at each stage, T stages
- The not-so-clever algorithm: brute force enumeration of all possible paths: $O(K^T)$
- We can do a lot better! The backtrace algorithm: $O(K^2T)$!

The core idea is to solve the problem recursively.

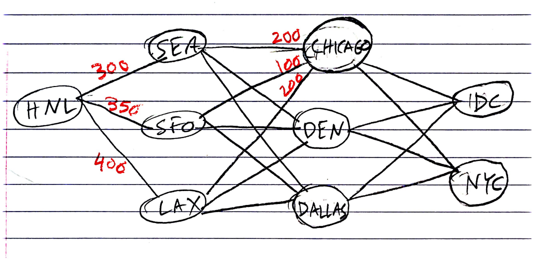
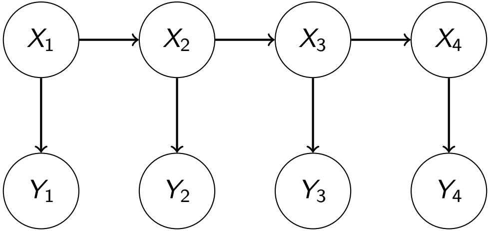


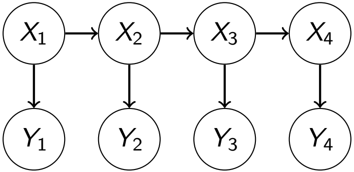
Figure: What is the cheapest way to get to CHICAGO?

- BACKTRACE: To solve the problem of finding the cheapest way to get to stage $t+1$, we only need to know the cheapest way to get to nodes in stage t.
- stage-to-stage computation is $O(K^2)$.
- There are T stages: the total computation is $O(K^2T)$

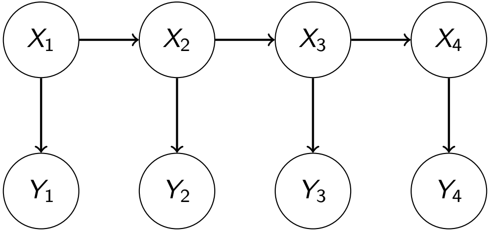


VITERBI BACKTRACE

$$\operatorname*{argmax}_{X_{1:T}} P(X_{1:T} \mid Y_{1:T})$$



- A set of K states in $X_t \in [K] = \{1, \ldots, K\}$.
- Transition probabilities
$$P(X_t | X_{t-1})$$
- A sequence of observations $Y_{1:T} = (Y_1, \ldots, Y_T)$ with $Y_t \in \mathcal{Y} = [L]$.
- A sequence of observation likelihoods, also called emission probabilities,
$$P(Y_t | X_t).$$
- An initial probability distribution over states denoted by $P(X_1)$.
- Given these, we know the joint probability $P(X_{1:T}, Y_{1:T})$:
$$P(X_{1:T}, Y_{1:T}) = \prod_{t=1}^{T} p(Y_t | X_t) P(X_t | X_{t-1})$$



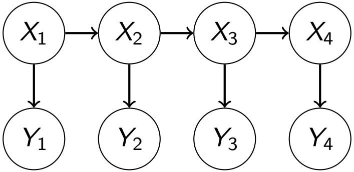
WARMUP: α-UPDATE ALGORITHM

FILTERING : $P(X_t | Y_{1:t})$

- Define $\alpha(X_t)$:
$$\alpha(X_t) = P(X_t, Y_{1:t})$$
- We can express $\alpha(X_t)$ in terms of $\alpha(X_{t-1})$:
$$\alpha(X_t) = P(Y_t|X_t) \sum_{X_{t-1}} \alpha(X_{t-1}) P(X_t|X_{t-1}), \quad t > 1$$
- The iteration starts at $\alpha(X_1) = P(Y_1|X_1)P(X_1)$

$$
\begin{aligned}
\alpha(X_t) &= P(X_t, Y_{1:t}) \\
&= \sum_{X_{t-1}} P(X_t, X_{t-1}, Y_{1:t-1}, Y_t) \\
&= \sum_{X_{t-1}} P(Y_t|X_t, X_{t-1}, Y_{1:t-1})P(X_t|X_{t-1}, Y_{1:t-1})P(X_{t-1}, Y_{1:t-1}) \\
&= \sum_{X_{t-1}} P(Y_t|X_t)P(X_t|X_{t-1})P(X_{t-1}, Y_{1:t-1}) \\
&= P(Y_t|X_t) \sum_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}, Y_{1:t-1}) \\
&= P(Y_t|X_t) \sum_{X_{t-1}} \alpha(X_{t-1})P(X_t|X_{t-1})
\end{aligned}
$$

VITERBI ALGORITHM I.1



- We are interested in the most likely sequence $X_{1:T}$ of $P(X_{1:T}|Y_{1:T})$:

$$\operatorname*{argmax}_{X_{1:T}} P(X_{1:T}|Y_{1:T}) = \operatorname*{argmax}_{X_{1:T}} P(X_{1:T}, Y_{1:T})$$

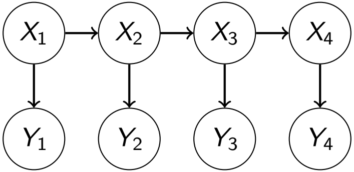
- Start with the following:

$$\max_{X_T} \prod_{t=1}^{T} P(Y_t|X_t)P(X_t|X_{t-1}) = \left(\prod_{t=1}^{T-1} P(Y_t|X_t)P(X_t|X_{t-1}) \right) \underbrace{\max_{X_T} P(Y_T|X_T)P(X_T|X_{T-1})}_{\mu(X_{T-1})}$$

- The "message" $\mu(X_{T-1})$ conveys information from the end of the chain to the penultimate timestep. We can continue recursively:

$$\mu(X_{t-1}) = \max_{X_t} P(Y_t|X_t)P(X_t|X_{t-1})\mu(X_t), \quad t = T, \ldots, 2$$

with $\mu(X_T) = 1$.



- Maximizing over $X_2, \ldots, X_T$ is “compressed” into the message $\mu(X_1)$ so that the most likely state X_1^* is given by

$$X_1^* = \underset{X_1}{\operatorname{argmax}}\, P(Y_1|X_1)P(X_1)\mu(X_1)$$

- Once computed, "BACKTRACKING" gives

$$X_t^* = \underset{X_t}{\operatorname{argmax}}\, P(Y_t|X_t)P(X_t|X_{t-1}^*)\mu(X_t), \quad t = 2, \ldots, T$$

- We could also solve the problem by passing "messages" in the forward fashion, starting with

$$\max_{X_1} \prod_{t=1}^{T} P(Y_t|X_t)P(X_t|X_{t-1})$$

$$= \underbrace{\max_{X_1} \overbrace{P(X_1)P(Y_1|X_1)}^{\nu(X_1)} P(X_2|X_1)P(Y_2|X_2)}_{\nu(X_2)} \left(\prod_{t=3}^{T} P(X_t|X_{t-1})P(Y_t|X_t)\right),$$

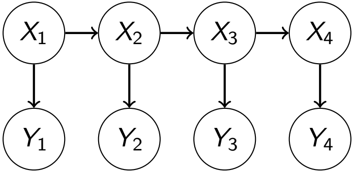
arriving at the recursion

$$\nu(X_t) = \max_{X_{t-1}} \nu(X_{t-1})P(X_t|X_{t-1}) \cdot P(Y_t|X_t), \qquad t = 2, \ldots, T.$$

- The recursion is initialized with

$$\nu(X_1) := P(X_1)P(Y_1|X_1).$$

VITERBI ALGORITHM II.2



- Now, BACKTRACKING starts with[2]

$$X_T^* = \underset{X_T}{\operatorname{argmax}} \, \nu(X_T).$$

- This is followed up by the recursion

$$X_{t-1}^* = \underset{X_{t-1}}{\operatorname{argmax}} \, \nu(X_{t-1}) P(X_t^* | X_{t-1}) P(Y_t | X_t^*), \quad t = T, \ldots, 2.$$

- This is the probabilistic form of our BACKTRACKING algorithm to find the cheapest flight!

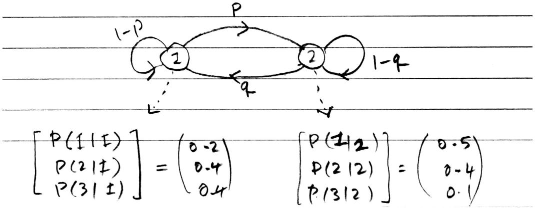
---

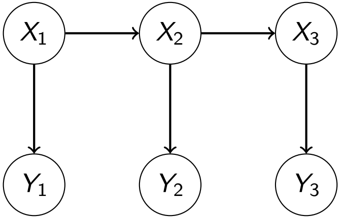
[2]Since T is the end of our iteration, we have

$$\nu(X_T^*) = \max_{X_{1:T}} P(X_{1:T} | Y_{1:T}).$$

EXAMPLE: $T = 3$

$$\underset{X_{1:3}}{\operatorname{argmax}}\, P(X_{1:3} | Y_{1:3}) = \underset{X_{1:3}}{\operatorname{argmax}}\, P(X_{1:3}, Y_{1:3})$$





$$P(X_{1:3}, Y_{1:3}) = P(X_1)P(Y_1|X_1)P(X_2|X_1)P(Y_2|X_2)P(X_3|X_2)P(Y_3|X_3)$$