Lecture 21. HMMs and Message Passing

COMP90051 Statistical Machine Learning

Semester 1, 2021
Lecturer: Trevor Cohn

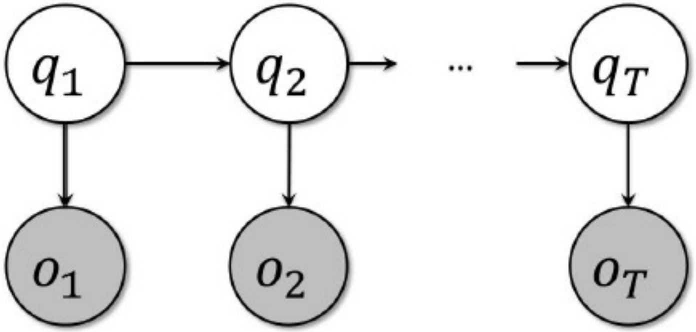
This lecture

- Hidden Markov models – detailed PGM case study
  - Brief recap of model
  - “Evaluation”: Forward-Background Algorithm = elimination
  - “Learning”: Baum Welch = MLE
  - “Decoding”: Viterbi = elimination variant with sum→max
- Message passing
  - Sum-product generalises elimination algorithm
  - Variants for ring operators, max-product for Viterbi
  - Factor graphs

Hidden Markov Models

Model of choice for sequential data. A form of clustering for discrete time series.

HMM Formulation



- Formulated as directed PGM
  - therefore joint expressed as

$$P(\mathbf{o}, \mathbf{q}) = P(q_1)P(o_1|q_1)\prod_{i=2}^{T} P(q_i|q_{i-1})P(o_i|q_i)$$

  - **bold** variables are shorthand for vector of *T* values

- Parameters (for *homogenous* HMM)

$A = \{a_{ij}\}$ transition probability matrix; $\forall i : \sum_j a_{ij} = 1$

$B = \{b_i(o_k)\}$ output probability matrix; $\forall i : \sum_k b_i(o_k) = 1$

$\Pi = \{\pi_i\}$ the initial state distribution; $\sum_i \pi_i = 1$

Fundamental HMM Tasks

HMM Task	PGM Task
Evaluation. Given an HMM μ and observation sequence $\boldsymbol{o}$, determine likelihood $\Pr(\boldsymbol{o}\|\mu)$	Probabilistic inference
Decoding. Given an HMM μ and observation sequence $\boldsymbol{o}$, determine most probable hidden state sequence $\boldsymbol{q}$	MAP point estimate
Learning. Given an observation sequence $\boldsymbol{o}$ and set of states, learn parameters A, B, Π	Statistical inference

"Evaluation" a.k.a. marginalisation

- Compute prob. of observations **o** by summing out **q**

$$P(\mathbf{o}|\mu) = \sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q}|\mu)$$
$$= \sum_{q_1} \sum_{q_2} \ldots \sum_{q_T} P(q_1)P(o_1|q_1)P(q_2|q_1)P(o_2|q_2)\ldots P(q_T|q_{T-1})P(o_T|q_T)$$

- Make this more efficient by moving the sums

$$P(\mathbf{o}|\mu) = \sum_{q_1} P(q_1)P(o_1|q_1) \sum_{q_2} P(q_2|q_1)P(o_2|q_2) \ldots \sum_{q_T} P(q_T|q_{T-1})P(o_T|q_T)$$

- Déjà vu? Maybe we could do var. elimination…

Elimination = Backward Algorithm

$$P(\mathbf{o}|\mu) = \sum_{q_1} P(q_1)P(o_1|q_1) \underbrace{\sum_{q_2} P(q_2|q_1)P(o_2|q_2) \ldots \underbrace{\sum_{q_T} P(q_T|q_{T-1})P(o_T|q_T)}_{m_{T\to T-1}(q_{T-1})}}_{m_{2\to 1}(q_1)}$$

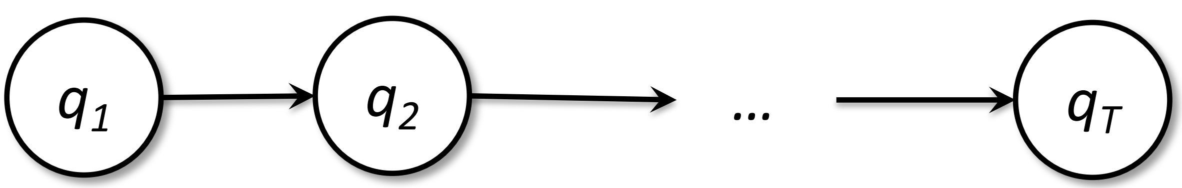
Eliminate q_T

…

Eliminate q_2

"Eliminate" q_1

$$P(\mathbf{o}|\mu) = \sum_{q_1} P(q_1)P(o_1|q_1)m_{2\to 1}(q_1)$$



Elimination = Forward Algorithm

$$P(\mathbf{o}|\mu) = \sum_{q_T} P(o_T|q_T) \underbrace{\sum_{q_{T-1}} P(q_T|q_{T-1})P(o_T|q_T) \ldots \underbrace{\sum_{q_1} P(q_2|q_1)P(q_1)P(o_1|q_1)}_{m_{1\to 2}(q_2)}}_{m_{T-1\to T}(q_T)}$$

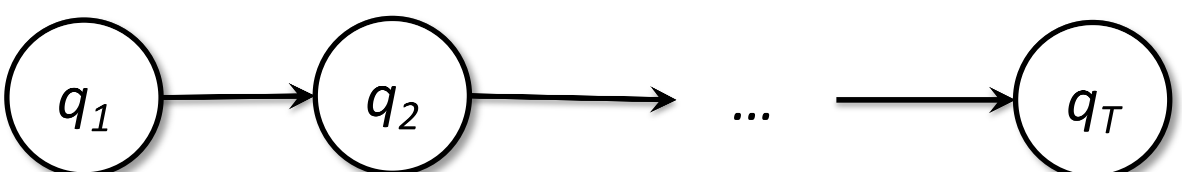
Eliminate q_1

...

Eliminate q_{T-1}

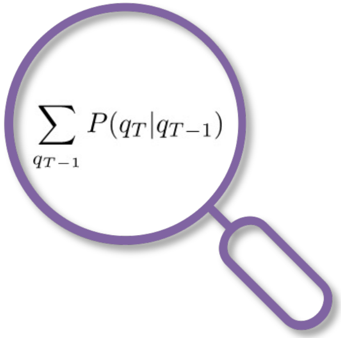
"Eliminate" q_T

$$P(\mathbf{o}|\mu) = \sum_{q_1} P(o_T|q_T) m_{T-1\to T}(q_T)$$



Variable elimination perspective

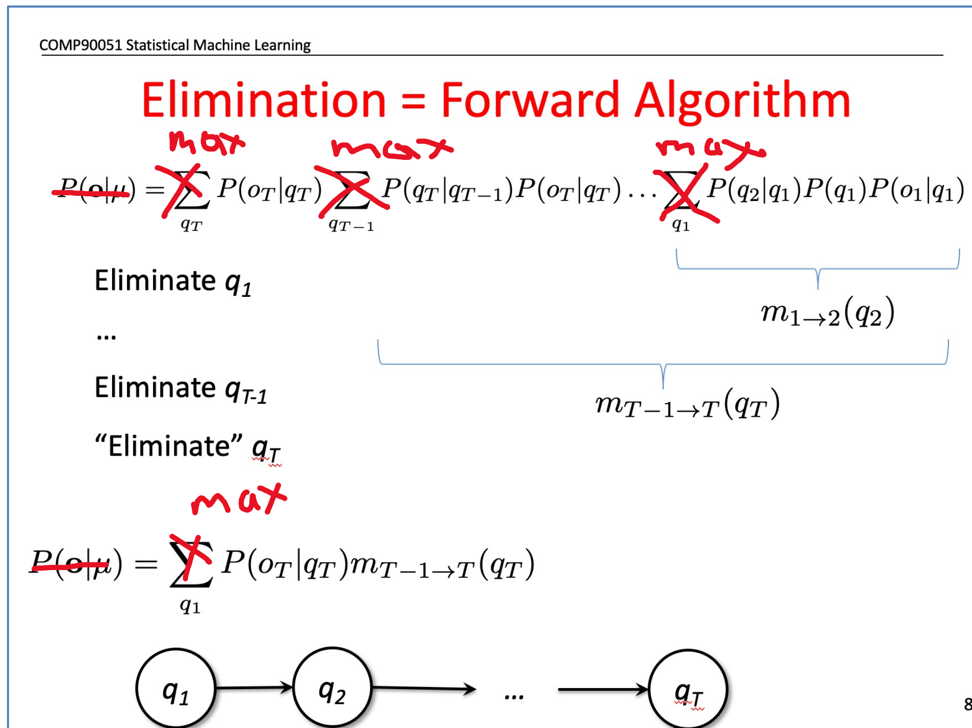
- Both algorithms are just *variable elimination* using different orderings
  - $q_T \ldots q_1 \rightarrow$ backward algorithm
  - $q_1 \ldots q_T \rightarrow$ forward algorithm
  - both have time complexity $O(TL^2)$ for L the label set size



- Can use either to compute P(**o**)
- Even though these are just instances of elimination, they pre-date general PGM inference.
  - E.g. called the "forward-backward algorithm"
  - Both directions useful in statistical inference (next)

Quick aside: Viterbi

- What happens if we switch sums for max?
  - Viola! Finds $\max_{\mathbf{q}} P(\mathbf{q}|\mathbf{o})$; and with some book-keeping can recover argmax
  - More on this later…



Mini Summary

- HMM
  - Powerful and versatile model
  - "Algorithms" for HMM just instances of PGM machinery
- Evaluation by Forward / Backward
  - Just elimination by two different orderings

Next time: Statistical inference (learning) example of EM

Statistical Inference (Learning)

- Learn parameters $\boldsymbol{\mu} = (A, B, \boldsymbol{\pi})$, given observation sequence $\mathbf{o}$
- Called "Baum Welch" algorithm which uses EM* to approximate MLE, $\text{argmax}_{\boldsymbol{\mu}} P(\mathbf{o}|\boldsymbol{\mu})$:
  1. initialise $\boldsymbol{\mu}^1$, let $j=1$
  2. compute expected marginal distributions $P(q_t|\mathbf{o}, \boldsymbol{\mu}^j)$ for all t; and $P(q_{t-1},q_t|\mathbf{o}, \boldsymbol{\mu}^j)$ for $t=2..T$
  3. fit model $\boldsymbol{\mu}^{j+1}$ based on expectations
  4. repeat from step 2, with $j=j+1$

E step

M step

- Expectations (2) computed using forward-backward

* Expectation-Maximisation (EM) is coming up

Forward-Backward for $P(q_i|\mathbf{o})$

- Forward-Backward gives: messages, $P(\mathbf{o})$
- Bayes rule: $P(q_i|\mathbf{o}) = \frac{P(q_i,\mathbf{o})}{P(\mathbf{o})}$
- Marginalisation: $P(q_i, \mathbf{o}) = \sum_{q_1,\dots,q_{i-1},q_{i+1},\dots,q_T} P(\mathbf{q}, \mathbf{o})$

$$= \left(\sum_{q_1,\dots,q_{i-1}} P(o_1,\dots,o_{i-1},q_1,\dots,q_i)\right) P(o_i|q_i) \left(\sum_{q_{i+1},\dots,q_T} P(o_{i+1},\dots,o_T,q_{i+1},\dots,q_T|q_i)\right)$$

$$= m_{i-1\to i}(q_i) P(o_i|q_i) m_{i+1\to i}(q_i)$$

$$\boxed{P(q_i|\mathbf{o}) = \frac{1}{P(\mathbf{o})} \underbrace{m_{i-1\to i}(q_i)}_{\text{forward}} P(o_i|q_i) \underbrace{m_{i+1\to i}(q_i)}_{\text{backward}}}$$

Forward-Backward for $P(q_{i-1}, q_i|\mathbf{o})$

- Similar pattern: $P(q_{i-1}, q_i|\mathbf{o}) = \frac{P(q_{i-1}, q_i, \mathbf{o})}{P(\mathbf{o})}$

- Marginalisation: $P(q_{i-1}, q_i, \mathbf{o}) = \sum_{q_1,\dots,q_{i-2},q_{i+1},\dots,q_T} P(\mathbf{q}, \mathbf{o})$

$$= \left(\sum_{q_1,\dots,q_{i-2}} P(o_1,\dots,o_{i-2},q_1,\dots,q_{i-1})\right) P(o_{i-1}|q_{i-1})P(q_i|q_{i-1})P(o_i|q_i) \left(\sum_{q_{i+1},\dots,q_T} P(o_{i+1},\dots,o_T,q_{i+1},\dots,q_T|q_i)\right)$$

$$= m_{i-2\to i-1}(q_{i-1}) P(o_{i-1}|q_{i-1})P(q_i|q_{i-1})P(o_i|q_i) m_{i+1\to i}(q_i)$$

$$\frac{1}{P(\mathbf{o})} m_{i-2\to i-1}(q_{i-1}) P(o_{i-1}|q_{i-1})P(q_i|q_{i-1})P(o_i|q_i) m_{i+1\to i}(q_i)$$

forward **backward**

Mini Summary

- Statistical inference for HMMs
  - “Just” learning or MLE as we’re frequentist here
  - Unobserved random variables means: EM (more later on)
  - Maximisation step: looks like MLE – nothing new
  - Expectation step: achieved by forward-backward messages
- “Baum-Welch” is the original name of this algorithm

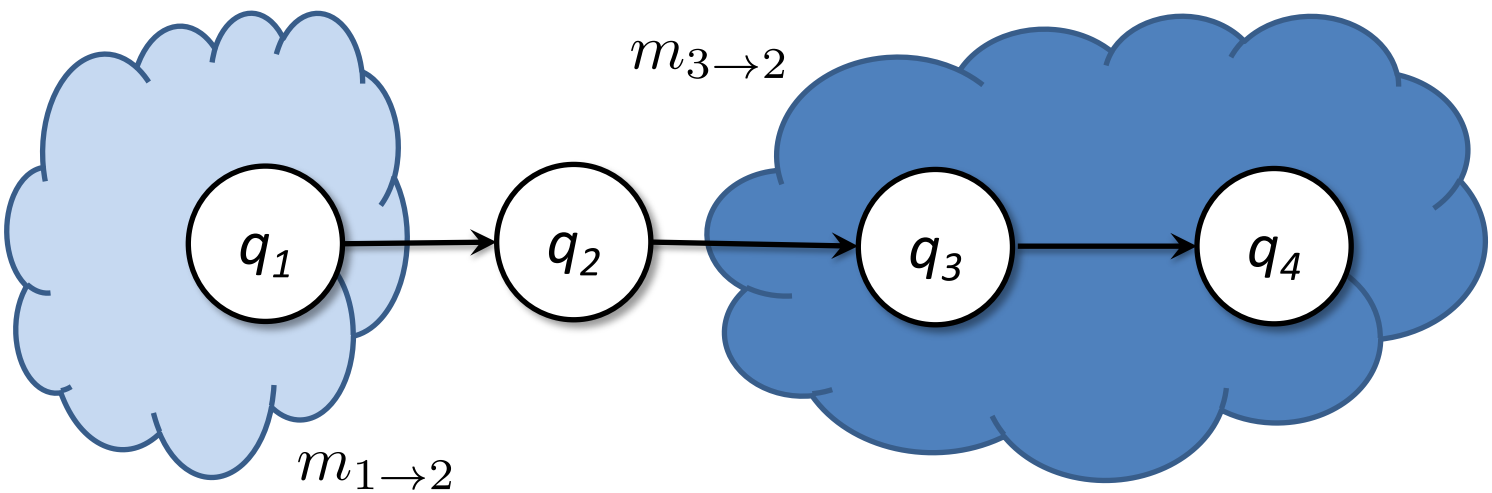
Next time: Message passing a little more generally

Message Passing

Sum-product algorithm for efficiently computing marginal distributions over trees. An extension of variable elimination algorithm.

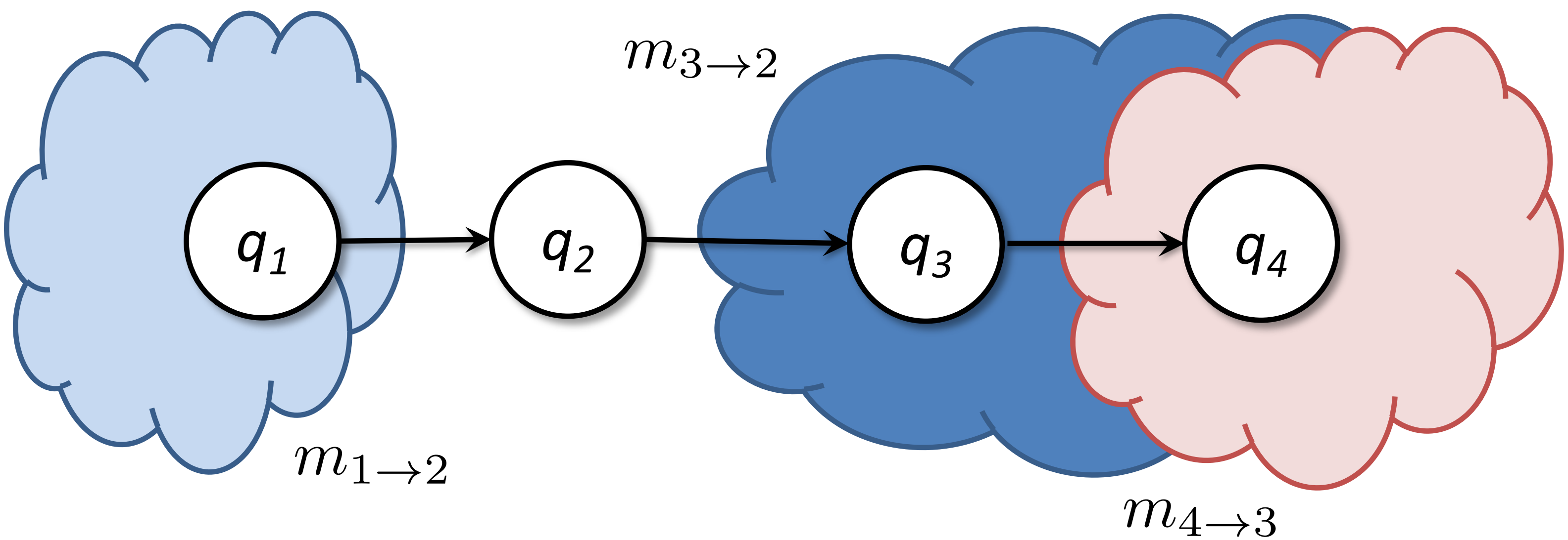
Inference as message passing

- Each *m* can be considered as a **message** which summarises the effect of the rest of the graph on the current node marginal.
  - *Inference = passing messages between all nodes*



Inference as message passing

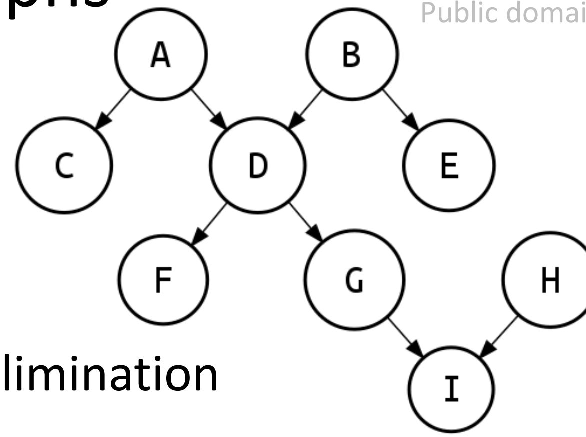
- Messages vector valued, i.e., function of target label
- Messages defined recursively: left to right, or right to left for the HMM



Sum-product algorithm

- Message passing in more general graphs
  - applies to chains, trees and poly-trees (D-PGMs with >1 parent)
  - ‘sum-product’ derives from:
    - **product** = product of incoming messages
    - **sum** = summing out the effect of rv(s) *aka* elimination

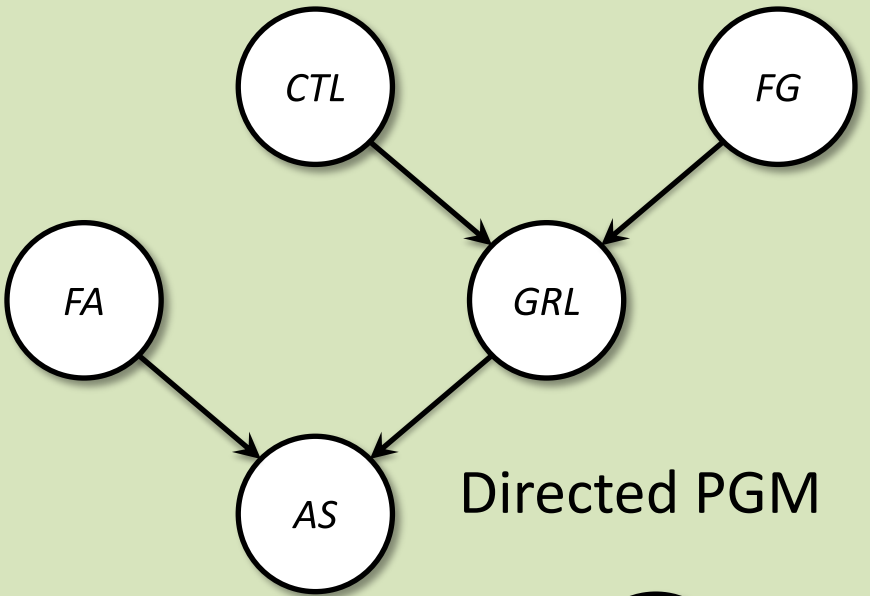
Public domain



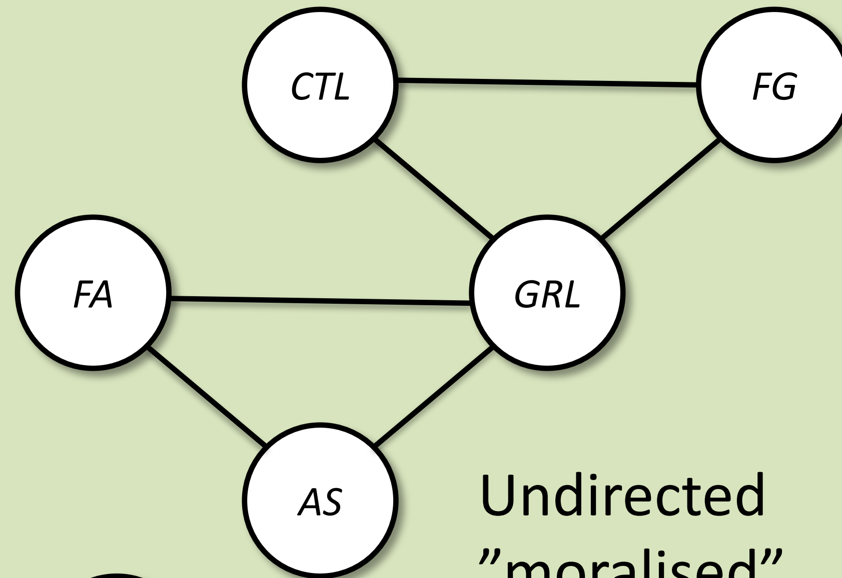
- Algorithm supports other operations (semi-rings*)
  - e.g., max-product, swapping **sum** for **max**
  - Viterbi algorithm is the max-product variant of forward algorithm, solves the $\text{argmax}_{\mathbf{q}} P(\mathbf{q}|\mathbf{o})$

** A ring is an algebraic structure generalizing addition/multiplication on reals. Semi-ring relaxes requirement of additive inverse.*

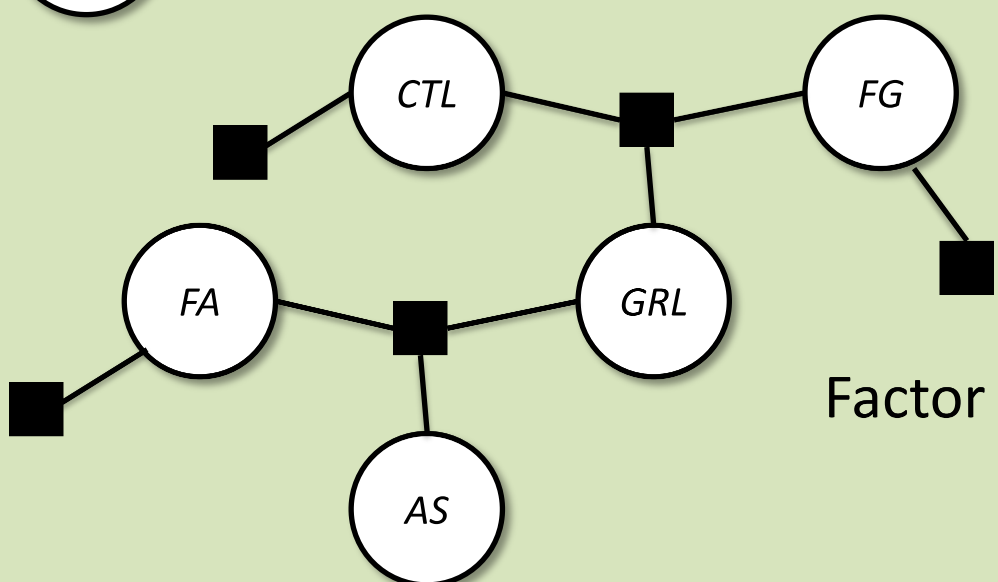
Application to Directed PGMS



Directed PGM



Undirected ”moralised” PGM

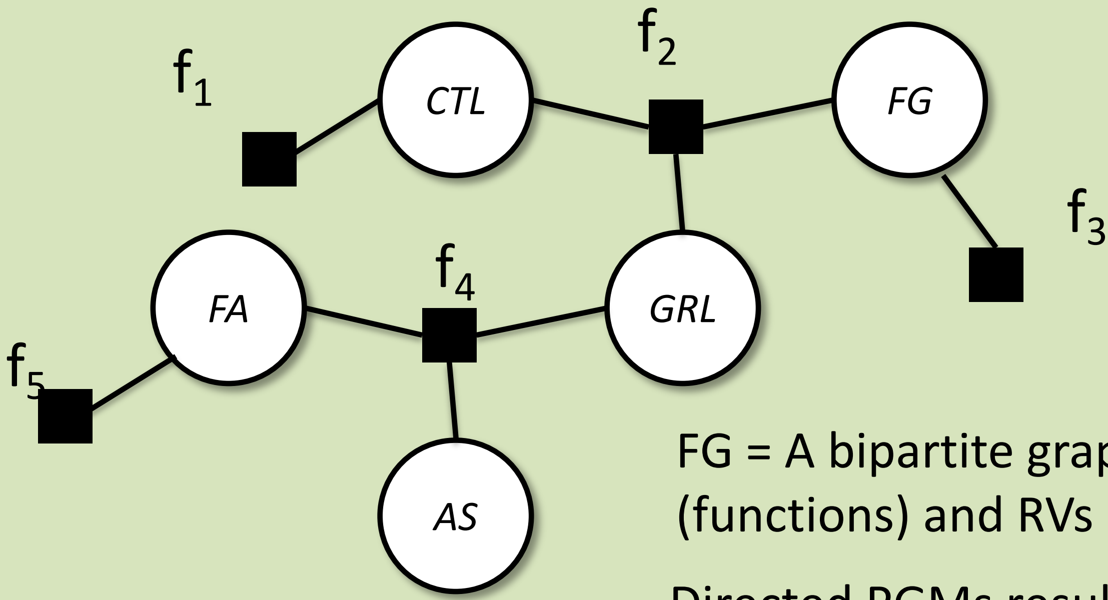


Factor graph

Factor graphs

$$f_1(CTL) = P(CTL)$$

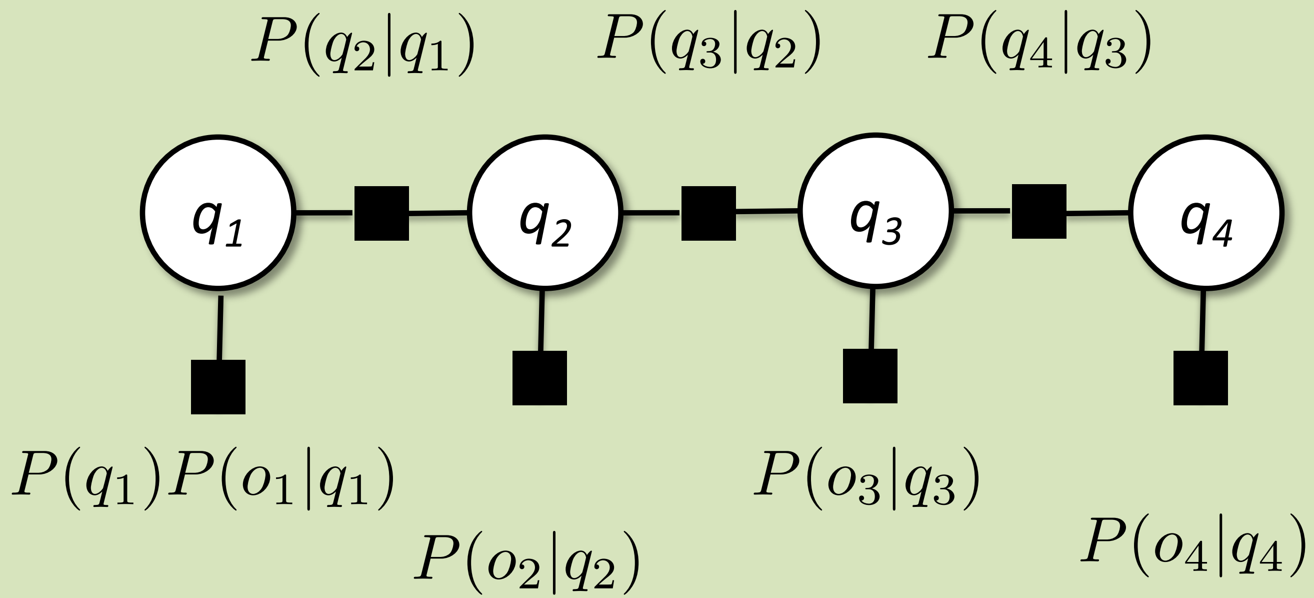
$$f_2(CTL, GRL, FG) = P(GRL|CTL, FG)$$



FG = A bipartite graph, with factors (functions) and RVs

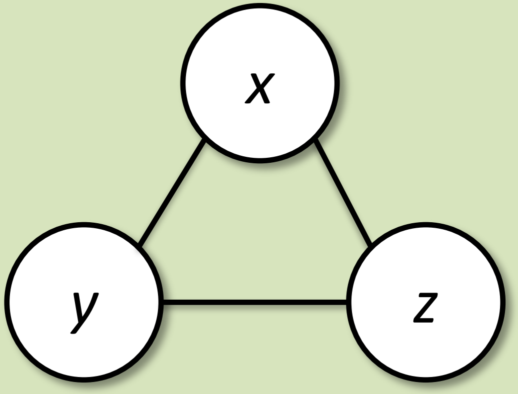
Directed PGMs result in tree-structured FG

Factor graph for the HMM



Effect of observed nodes incorporated into unary factors

Advantage of Factor Graphs



- Factorisation is a central idea
- D-PGMs and U-PGMs not able to fully represent arbitrary factorisations of joints

$$p(x,y,z) \propto \varphi(x,y)\varphi(y,z)\varphi(z,x)$$
$$p(x,y,z) \propto \varphi(x,y,z)$$

- Better representation of factorisations has advantages; factor graphs are general.

Sum-Product over Factor Graphs

- Two types of messages :
  - between factors and RVs; and between RVs and factors
  - they summarise a complete sub-graph
- E.g.,

$$m_{f_2 \to GRL}(GRL) = \sum_{CTL} \sum_{FG} f_2(GRL, CTL, FG) m_{CTL \to f_2}(CTL) m_{FG \to f_2}(FG)$$

- Structure inference as “gather-and-distribute”
  - gather messages from leaves of tree towards root
  - then propagate message back down from root to leaves

Summary

- HMMs as example PGMs
  - formulation as PGM
  - independence assumptions
  - probabilistic inference using forward-backward
  - statistical inference using expectation-maximization
  - decoding as max-product
- Message passing: general inference method for U-PGMs
  - sum-product & max-product
  - factor graphs

Next time: Gaussian mixture models and EM