6.867 Machine learning and neural networks

Tommi Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Lecture 23: Exact inference

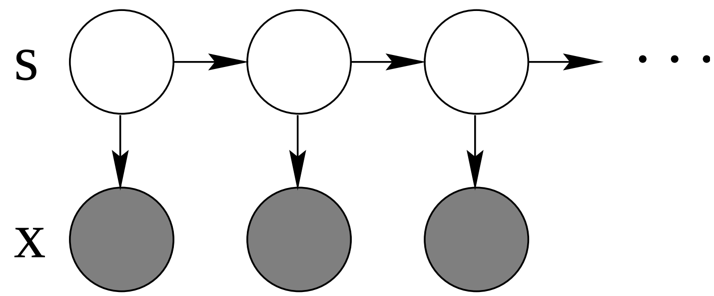
Topics

- Exact inference
  - Basic concepts
  - General algorithm

Nature of probabilistic inference

- Example: a hidden Markov model

$$P(s_0, x_0, \ldots, s_n, x_n) = P_0(s_0)\, P_o(x_0|s_0)\, P_1(s_1|s_0) \cdots$$



- Given the observation sequence $x_0^*, \ldots, x_n^*$, all the information about the associated hidden states is already contained in the joint probability distribution

$$P(s_0, x_0^*, \ldots, s_n, x_n^*)$$

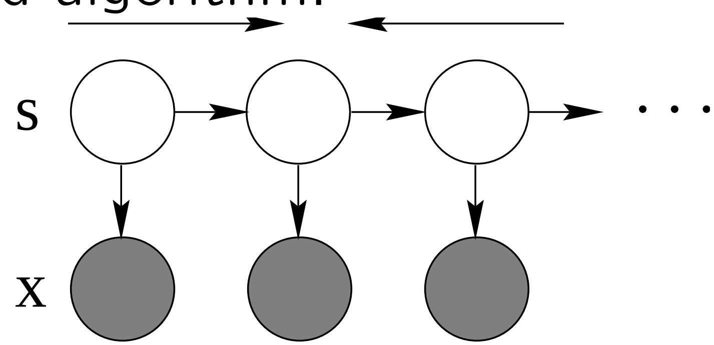
- What's left to do?

Nature of probabilistic inference

- We have to *explicate* the relevant information

  This involves propagation of information across the graph model

- Forward-backward algorithm:

  Forward step: information from the past about the current state

  Backward step: information from future observations about the current state

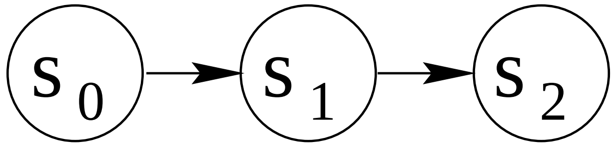
- We want analogous computations for more general graph models

Objective

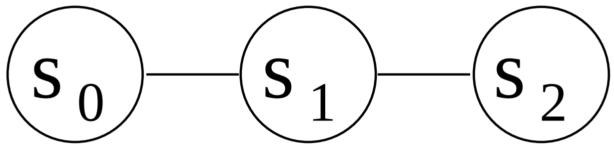
- Our objectives:
  1. Explicate relevant information
  2. Ensure locality of information
- For this we need
  1. to define an appropriate data structure (junction tree) where these calculations can be made
  2. to specify how information is propagated in such structures

Simple Markov chain example

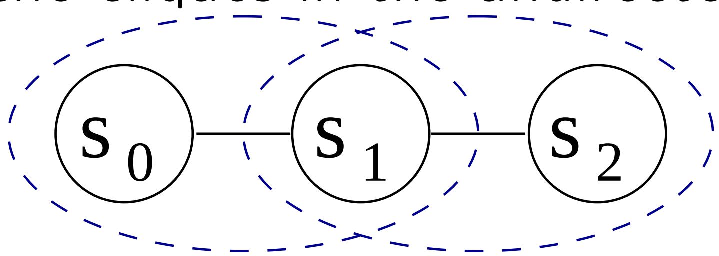
- Take a simple Markov chain



- We can replace the directed edges with undirected edges without affecting the graph semantics (or the underlying probability model)

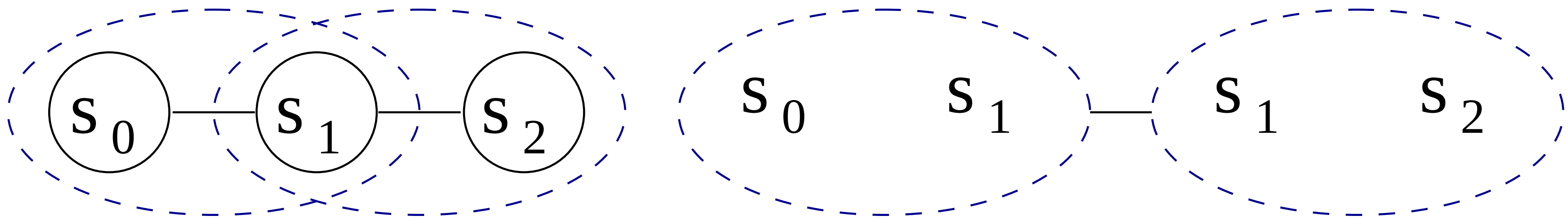


- We can identify the cliques in the undirected graph

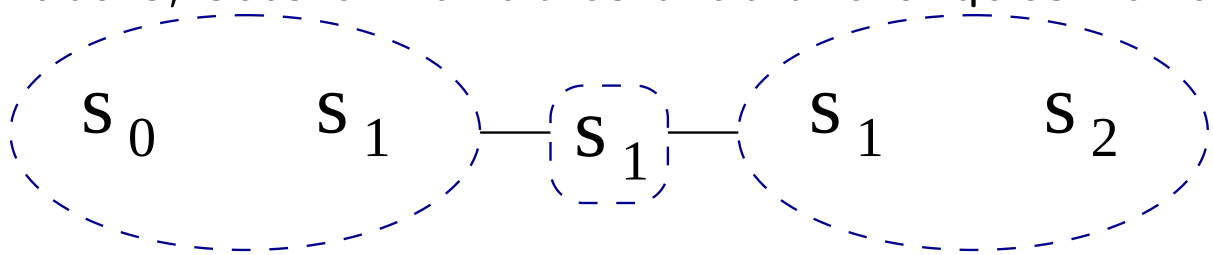


Markov chain example: clustering of nodes

- The cliques can be connected to define a hyper-graph (two cliques are connected if they have variables in common)

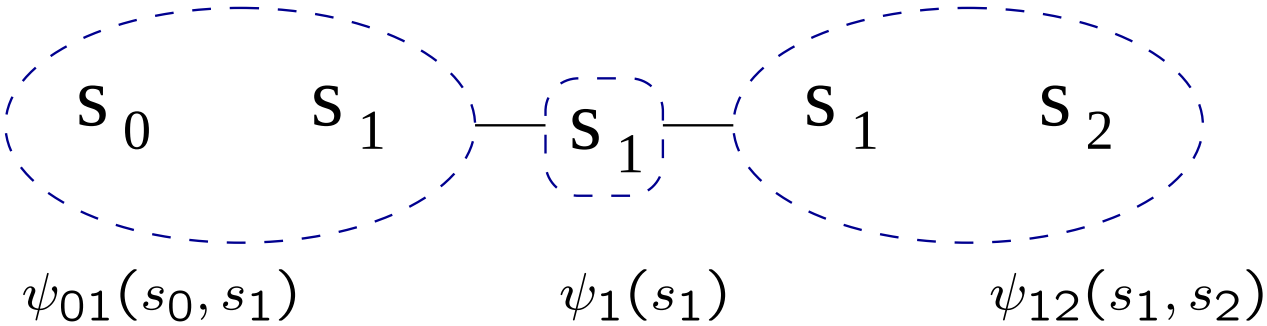


- Finally, we can explicate the overlap between the cliques by defining *separators*, sets of variables that the cliques have in common



- This is known as a *junction tree*

Probabilities and junction tree



- The joint distribution over the markov chain can be defined by associating potential functions with the cliques (and the separator)

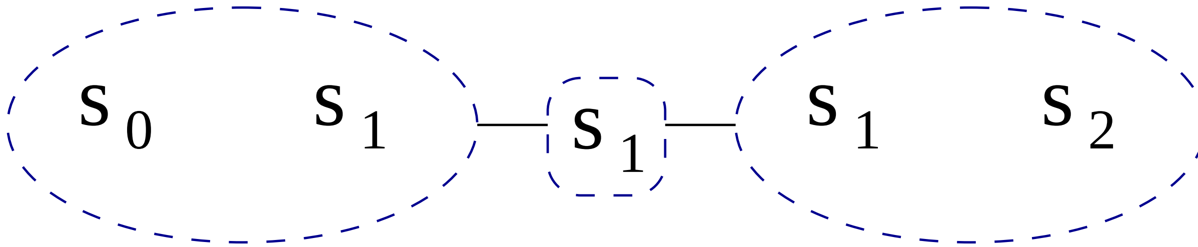
$$P(s_0, s_1, s_2) = \frac{P(s_0, s_1)\, P(s_1, s_2)}{P(s_1)} = \frac{\psi_{01}(s_0, s_1)\, \psi_{12}(s_1, s_2)}{\psi_1(s_1)}$$

- We assume here that initially

$$\begin{aligned} \psi_{01}(s_0, s_1) &\propto P(s_0, s_1) \\ \psi_{12}(s_1, s_2) &\propto P(s_1, s_2) \\ \psi_1(s_1) &\propto P(s_1) \end{aligned}$$

so that the information (marginal probabilities) about the variables in the cliques resides *locally* and is *explicit*

Evidence in a junction tree



$\psi_{01}(s_0, s_1)$ $\psi_1(s_1)$ $\psi_{12}(s_1, s_2)$

- When we acquire evidence about the values of the variables, the relevant information need not be local nor explicit any more

$$P(s_0^*, s_1, s_2) = \frac{P(s_0^*, s_1)\, P(s_1, s_2)}{P(s_1)} = \frac{\psi_{01}(s_0^*, s_1)\, \psi_{12}(s_1, s_2)}{\psi_1(s_1)}$$

  Here we assume that we have observed s_0^*

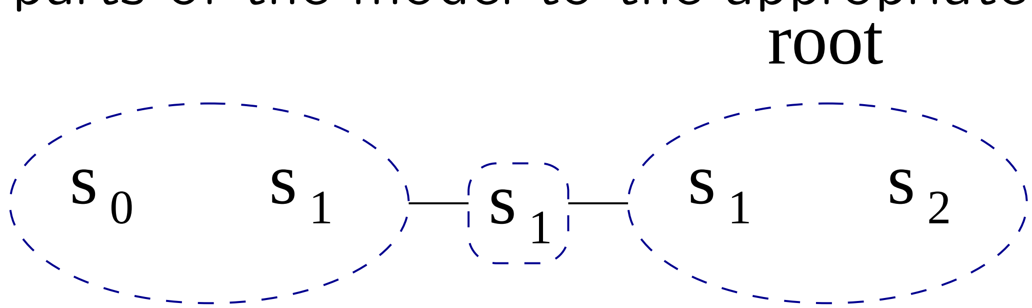
- To incorporate this evidence, we multiply the corresponding clique potential with an indicator function

$$\psi_{01}(s_0, s_1) \leftarrow \psi_{01}(s_0, s_1)\, \delta(s_0, s_0^*)$$

  where $\delta(s_0, s_0^*) = 1$ if $s_0 = s_0^*$ and zero otherwise.
  (potential is zero if we contradict the evidence)

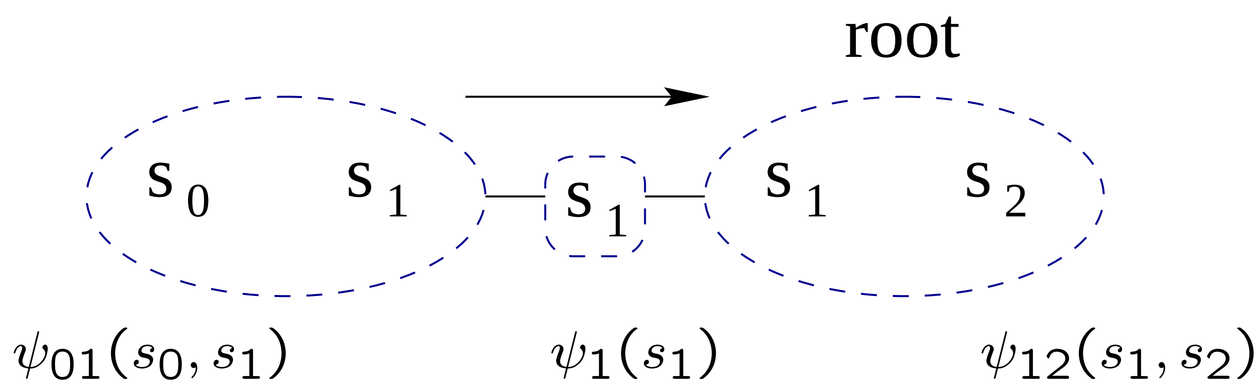
Junction tree algorithm

- To achieve locality, we must propagate the relevant information from other parts of the model to the appropriate cliques



- The junction tree (clustering) algorithm
  1. Pick a root node
  2. Collect information towards the root
  3. Distribute information away from the root
- Why do we need the two passes?

Collect operation



- Compute a target for the separator

$$\psi_1'(s_1) \leftarrow \sum_{s_0} \psi_{01}(s_0, s_1) \quad (= P(s_0^*, s_1)\,)$$

- Update the collecting clique (root in our case) based on this *new* information

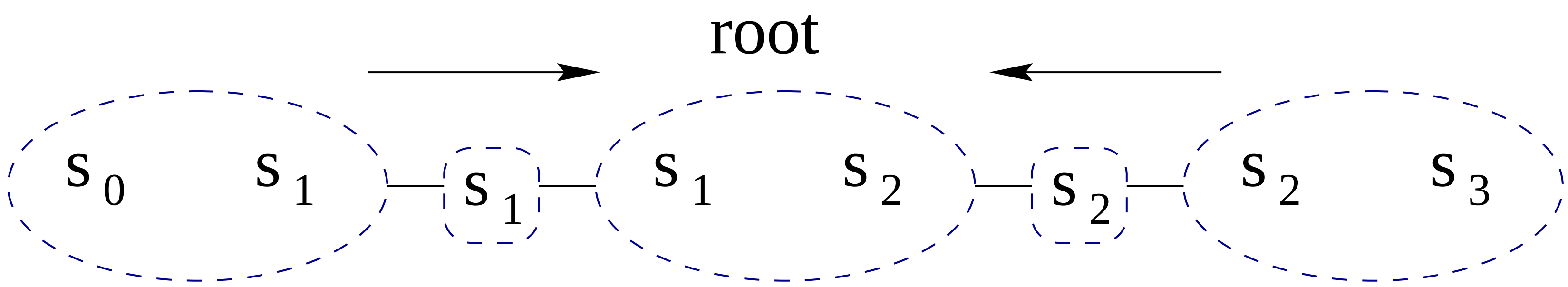
$$\psi_{12}(s_1, s_2) \leftarrow \frac{\psi_1'(s_1)}{\psi_1(s_1)} \psi_{12}(s_1, s_2)$$

  (nothing would change if there were no evidence)

- Update the separator

$$\psi_1(s_1) \leftarrow \psi_1'(s_1)$$

General collect operation

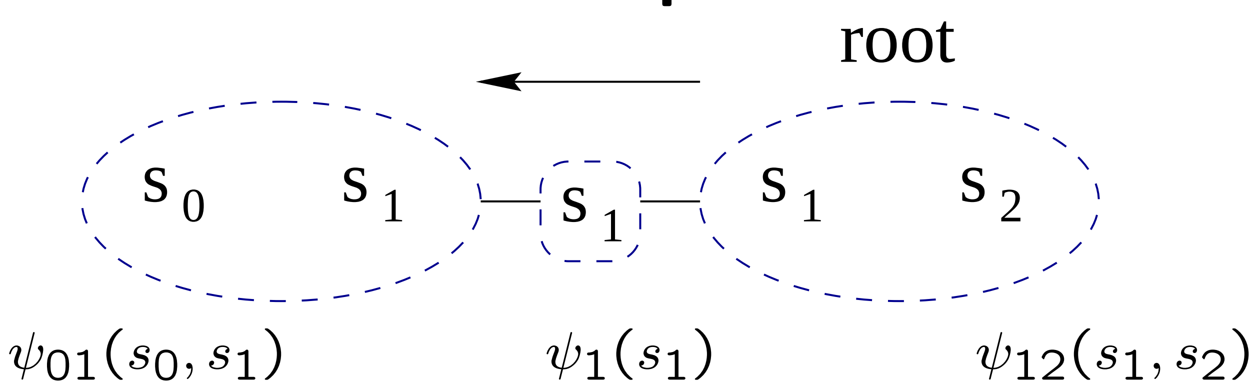


- When collecting from multiple neighbors, we must update the cliques based on all the incoming information

$$\psi_{12}(s_1, s_2) \leftarrow \frac{\psi_1'(s_1)}{\psi_1(s_1)} \frac{\psi_2'(s_2)}{\psi_2(s_2)} \psi_{12}(s_1, s_2)$$

(again nothing would change if there were no evidence)

Distribute operation



- Compute a target value for the separator (now in the other direction)

$$\psi_1'(s_1) \leftarrow \sum_{s_2} \psi_{12}(s_1, s_2)$$

- Update the receiving clique based on this *new* information

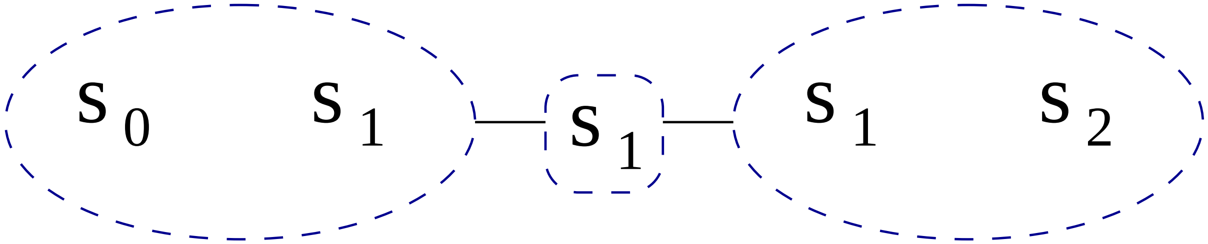
$$\psi_{01}(s_0, s_1) \leftarrow \frac{\psi_1'(s_1)}{\psi_1(s_1)} \psi_{01}(s_0, s_1)$$

(note that in our case there's is no new information; update doesn't change anything)

- Finally, we update the separator

$$\psi_1(s_1) \leftarrow \psi_1'(s_1)$$

Junction tree



$\psi_{01}(s_0, s_1)$ $\psi_1(s_1)$ $\psi_{12}(s_1, s_2)$

- After both propagation operations, the relevant information is again stored locally

$$\begin{aligned}
\psi_{01}(s_0, s_1) &\propto P(s_0, s_1 | \text{evidence}) \\
\psi_1(s_1) &\propto P(s_1 | \text{evidence}) \\
\psi_{12}(s_1, s_2) &\propto P(s_1, s_2 | \text{evidence})
\end{aligned}$$