Machine learning: lecture 22

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

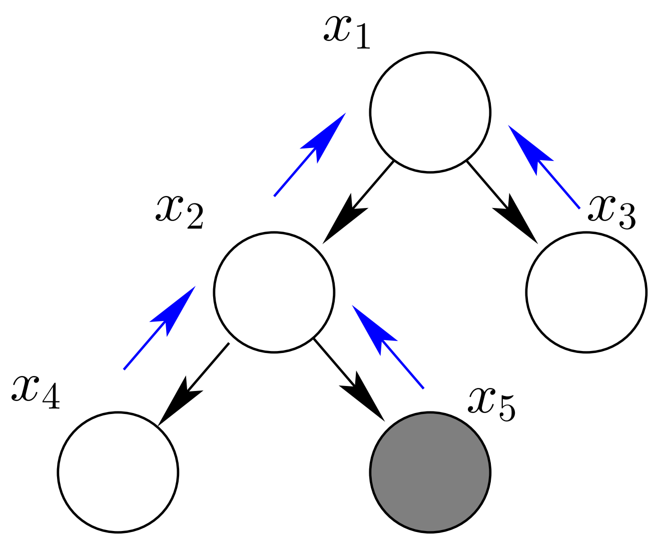
Topics

- Exact inference: belief propagation
  – basic idea, messsages
  – marginals and pairwise marginals
  – example

Belief propagation: preliminaries

- Belief propagation operates by sending messages between nearby variables in the graphical model

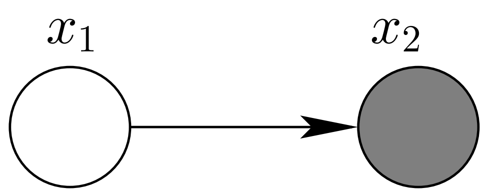
  The messages contain all the information (evidence) necessary to locally evaluate the posterior marginals for each variable



- Key issues: locality of information, use of the graph structure

Belief propagation: preliminaries

- To better formulate the message passing algorithm, we use the following common notation for the underlying probability distribution and the evidence about the values of the variables



$P(x_1, x_2) = P(x_1)P(x_2|x_1)$

Observation: $x_2 = \hat{x}_2$

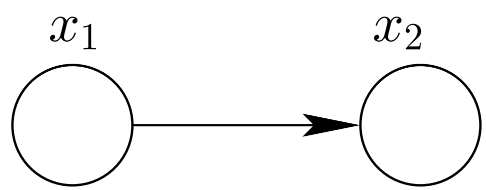
$$\psi_1(x_1) = P(x_1)$$
$$\psi_{12}(x_1, x_2) = P(x_2|x_1)$$
$$\psi_2(x_2) = \delta(x_2, \hat{x}_2)$$

- The potential functions (tables of non-negative numbers) are chosen so that

$$P(x_1, x_2, \text{observed data}) = \psi_1(x_1)\psi_{12}(x_1, x_2)\psi_x(x_2)$$

(the assignment of probabilities/evidence to potential functions is not unique)

Belief propagation: simple example



$$P(x_1, x_2) = P(x_1)P(x_2|x_1)$$

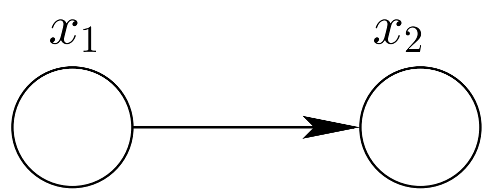
$$\psi_1(x_1) = P(x_1)$$
$$\psi_{12}(x_1, x_2) = P(x_2|x_1)$$
$$\psi_2(x_2) = 1$$

- We can evaluate marginals by sending messages:

$$\begin{aligned} P(x_1) &= \sum_{x_2} P(x_1, x_2) \\ &= \sum_{x_2} \psi_1(x_1)\psi_{12}(x_1, x_2)\psi_2(x_2) \\ &= \psi_1(x_1) \sum_{x_2} \psi_{12}(x_1, x_2)\psi_2(x_2) \\ &= \psi_1(x_1)\, m_{2\to 1}(x_1) \end{aligned}$$

where $m_{2\to 1}(x_1)$ is a table (message) that x_2 needs to send to x_1

Belief propagation: simple example cont'd



$$P(x_1, x_2) = P(x_1)P(x_2|x_1)$$

$$\psi_1(x_1) = P(x_1)$$
$$\psi_{12}(x_1, x_2) = P(x_2|x_1)$$
$$\psi_2(x_2) = 1$$

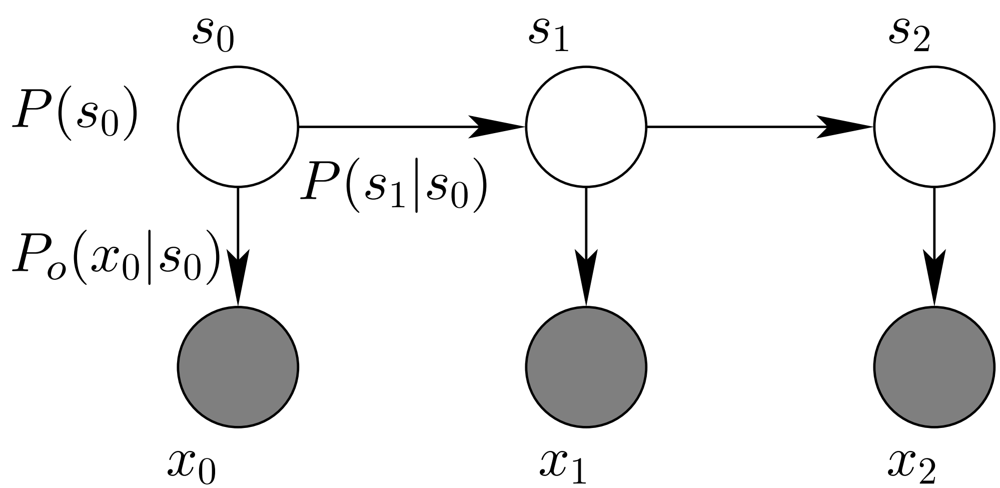
- We can analogously evaluate the marginal over x_2

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} \psi_1(x_1)\psi_{12}(x_1, x_2)\psi_2(x_2) \\ &= \psi_2(x_2) \sum_{x_1} \psi_{12}(x_1, x_2)\psi_1(x_1) \\ &= \psi_2(x_2)\, m_{1\to 2}(x_2) \end{aligned}$$

where $m_{1\to 2}(x_2)$ is a message that x_1 sends to x_2

Belief propagation: hidden Markov models

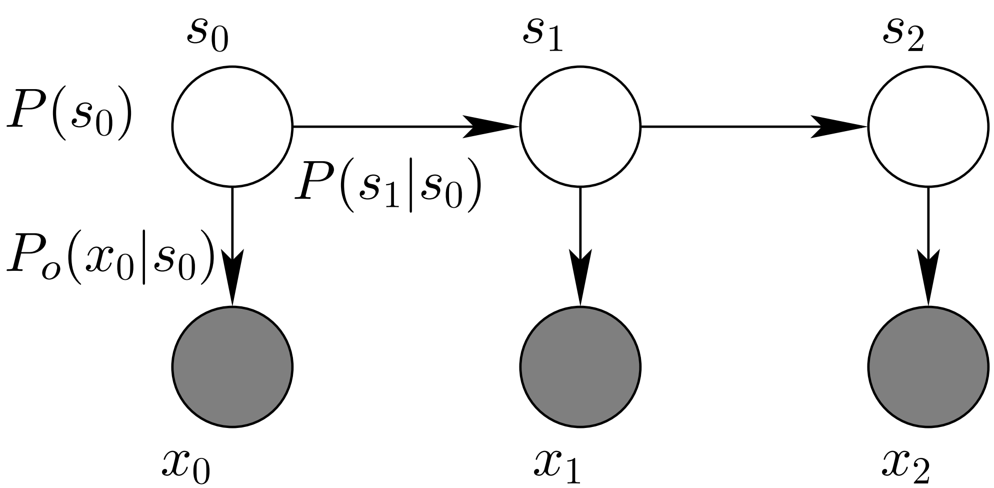
- Suppose we have the following hidden Markov model over three time steps, where the hidden (homogeneous) Markov chain has m states



- We are interested in the posterior probabilities over the hidden states s_t
- We formulate a message passing algorithm for this that operates analogously to the forward-backward algorithm

Hidden Markov models: notation

- We will first transform the probabilities and the evidence into potential functions



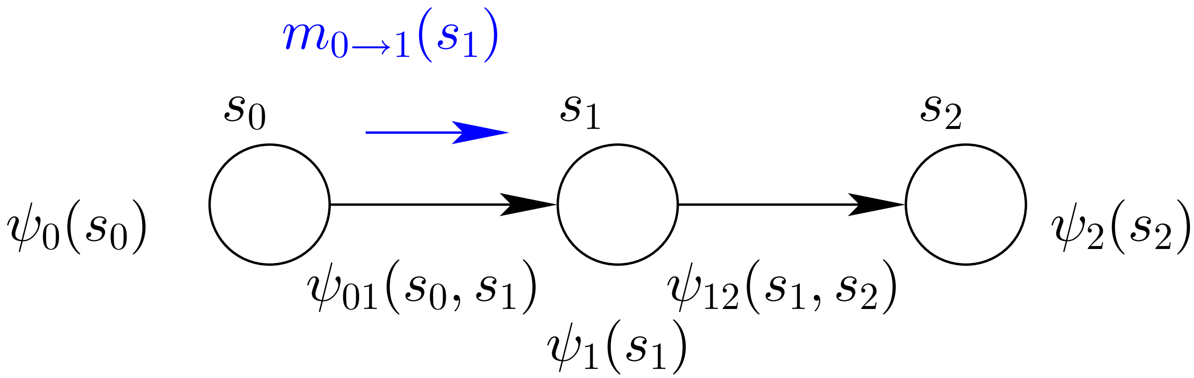
$$\psi_0(s_0) = P_o(x_0|s_0)P(s_0)$$
$$\psi_1(s_1) = P_o(x_1|s_1)$$
$$\psi_2(s_2) = P_o(x_2|s_2)$$

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

The distribution over the hidden states is given by

$$\begin{aligned} P(s_0, s_1, s_2, \text{data}) &= \psi_0(s_0)\psi_{01}(s_0, s_1) \cdot \\ &\quad \psi_1(s_1)\psi_{12}(s_1, s_2)\psi_2(s_2) \end{aligned}$$

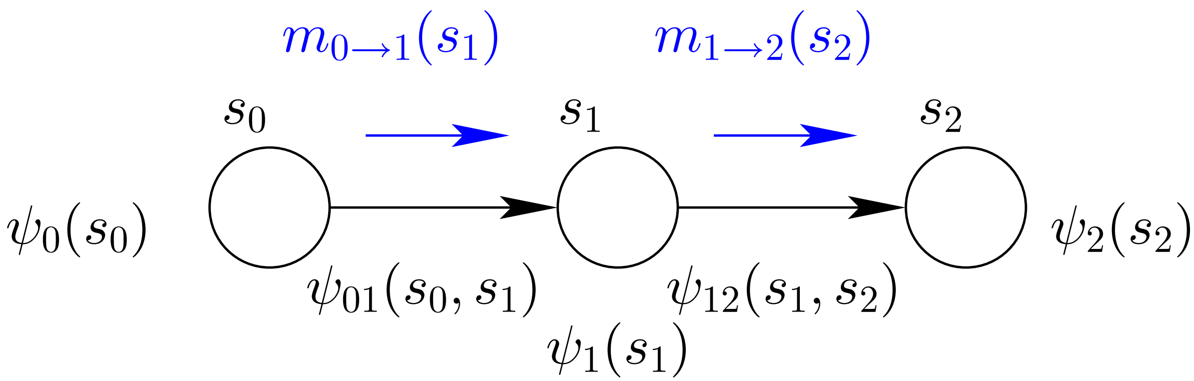
Hidden Markov models: messages



- To evaluate the marginal posterior over s_2 we need to sum over the possible values of s_0 and s_1. First,

$$\begin{aligned} P(s_1, s_2, \text{data}) &= \sum_{s_0} P(s_0, s_1, s_2, \text{data}) \\ &= \sum_{s_0} \psi_0(s_0)\psi_{01}(s_0, s_1)\,\psi_1(s_1)\psi_{12}(s_1, s_2)\psi_2(s_2) \\ &= m_{0\to 1}(s_1)\,\psi_1(s_1)\psi_{12}(s_1, s_2)\psi_2(s_2) \end{aligned}$$

Hidden Markov models: messages cont'd



- Finally

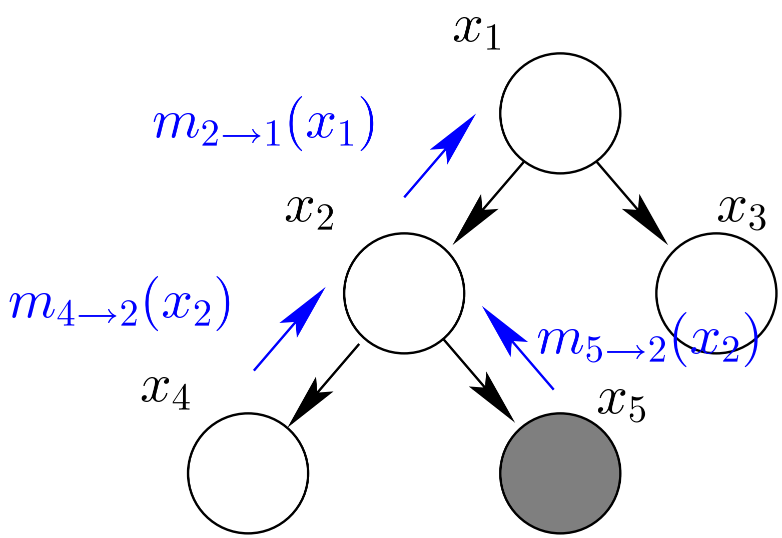
$$\begin{aligned} P(s_2, \text{data}) &= \sum_{s_1} P(s_1, s_2, \text{data}) \\ &= \sum_{s_0} m_{0\to1}(s_1)\psi_1(s_1)\psi_{12}(s_1, s_2)\psi_2(s_2) \\ &= m_{1\to2}(s_2)\,\psi_2(s_2) \end{aligned}$$

- To evaluate the other posterior marginals we would have to send analogous messages backwards in time

Belief propagation for trees

- In the more general case we have to combine messages from different branches of the tree (independent sources of information)

  Example:



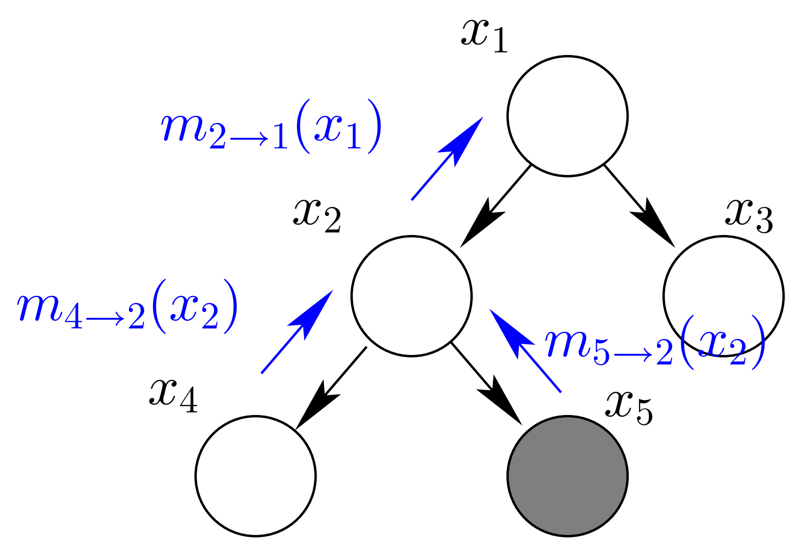
$$m_{2\to1}(x_1) \;=\; \sum_{x_2} \psi_{12}(x_1,x_2)\psi_2(x_2)\, m_{4\to2}(x_2) m_{5\to2}(x_2)$$

Belief propagation cont'd

- We can now define in general how each node in the graph should communicate with its upstream and downstream nodes (neighbors).

  A message passing scheme:

  1. Initialize messages to 1.
  2. Message (information) that j sends to i is



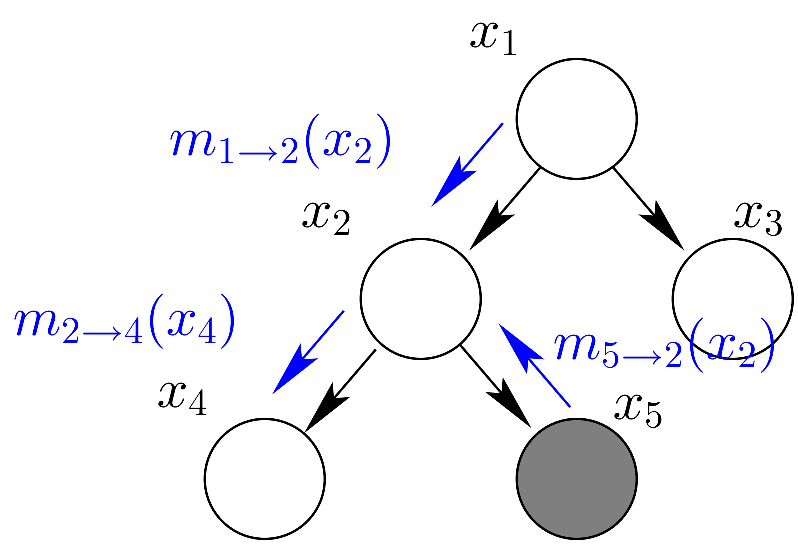
$$m_{j\to i}(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j)\psi_j(x_j) \prod_{l \neq i} m_{l\to j}(x_j)$$

where the product is over the neighbors of j not including i who the message is sent to

Belief propagation cont'd

- The messages are sent both upstream and downstream in the same manner

$$m_{j\to i}(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j)\psi_j(x_j) \prod_{l \neq i} m_{l\to j}(x_j)$$



Belief propagation cont'd

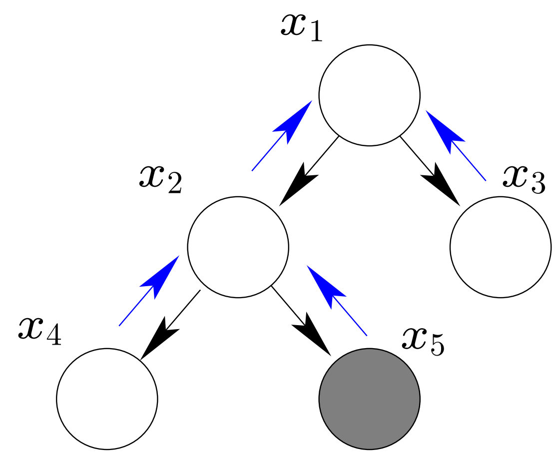
$$m_{j\to i}(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j)\psi_j(x_j) \prod_{l\neq i} m_{l\to j}(x_j)$$

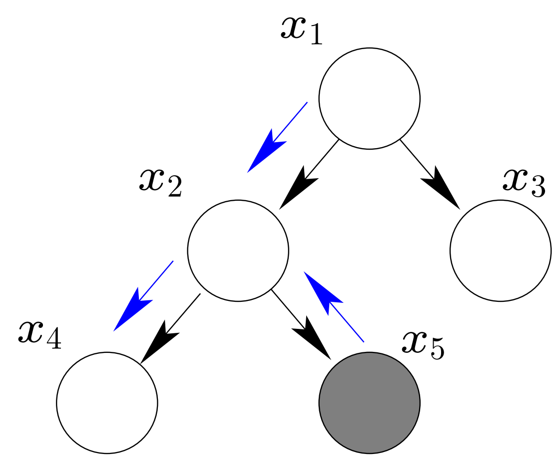
- This message passing algorithm is guaranteed to converge when the graph is a tree and

  $$P(x_i, \text{data}) = \psi_i(x_i) \prod_j m_{j\to i}(x_i)$$

  (the product is over neighbors of i)

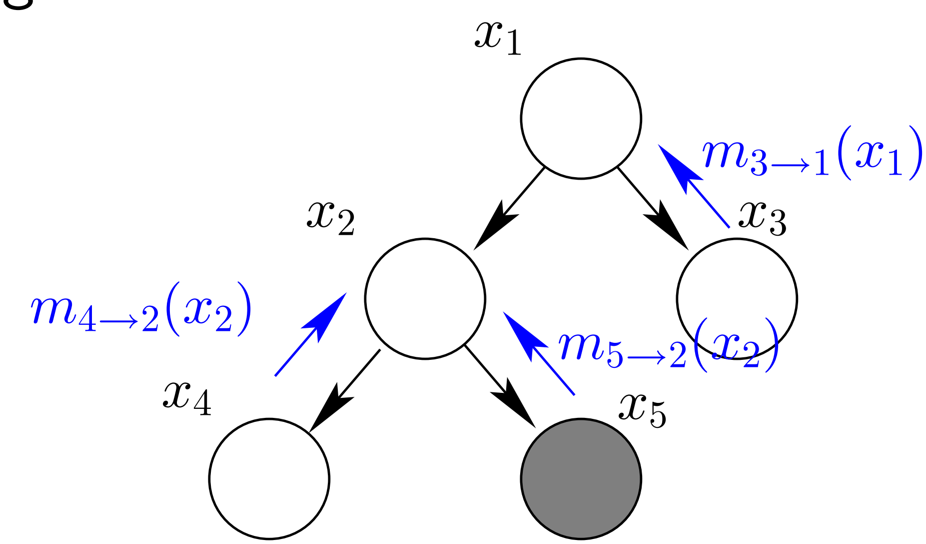
- The messages can be updated synchronously, asynchronously, or via a reasonable schedule (e.g., leaf-root-leaf)





Belief propagation cont'd

- We can also easily construct pairwise posterior marginals for neighboring variables in the graph from the converged messages:



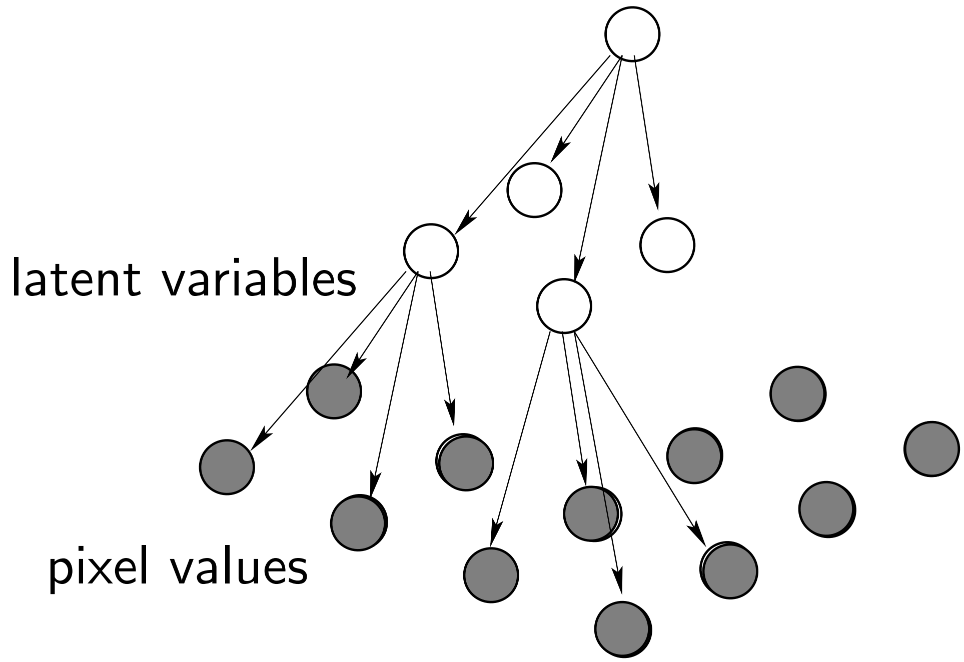
$$P(x_1, x_2, \text{data}) =$$

$$m_{3\to1}(x_1)\,\psi_1(x_1)\psi_{12}(x_1,x_2)\psi_2(x_2)\,m_{4\to2}(x_2)m_{5\to2}(x_2)$$

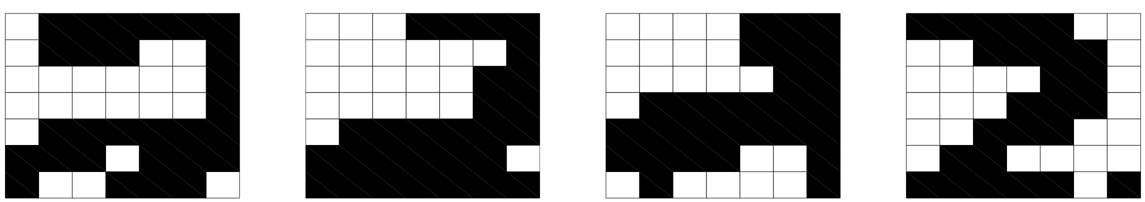
- Evaluation of non-neighbor pairwise marginals is somewhat harder

Example

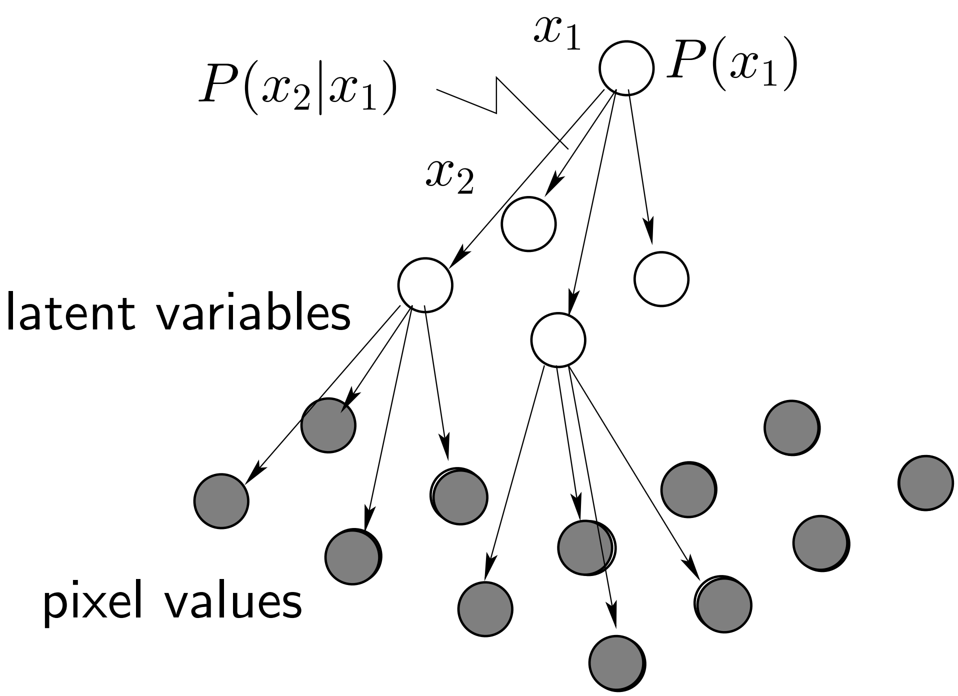
- A latent tree model

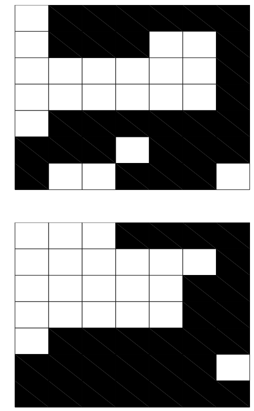


- We'd like to be able to estimate such latent variable models from a set of pixel images



Estimation via EM

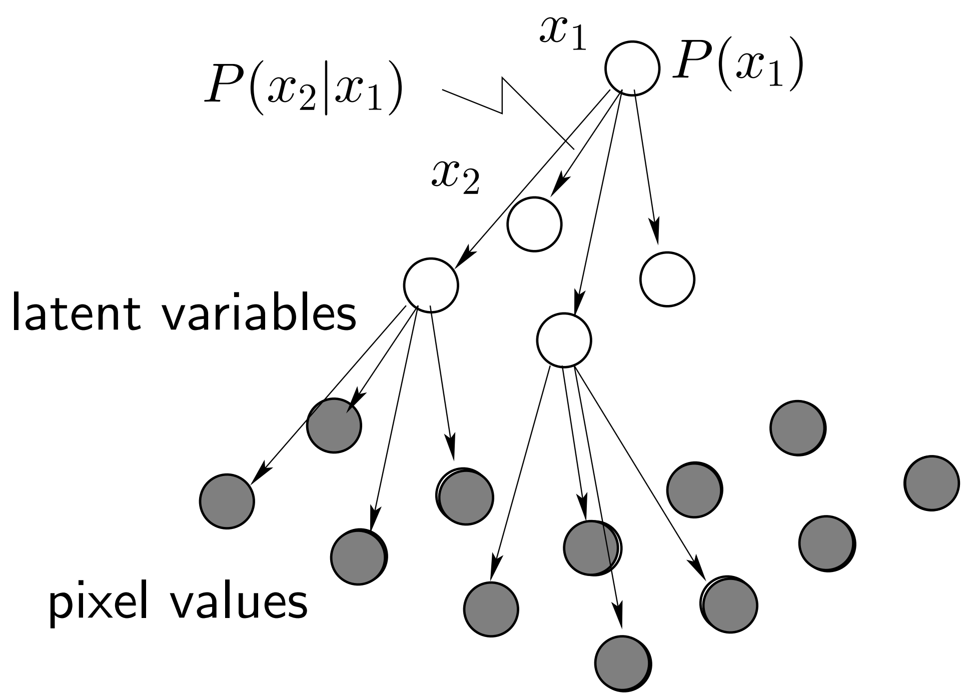




- We can try to find the maximum likelihood setting of the parameters via the EM algorithm

  E-step: we have to evaluate pairwise posterior marginals $P(x_i, x_j|\text{image})$ for neighboring nodes in the tree, separately for each image

Estimation via EM cont'd



M-step: we can update the conditional probabilities $P(x_i|x_j)$ in the tree by normalizing the soft posterior "counts"

$$\hat{n}(x_i, x_j) = \sum_t P(x_i, x_j|\text{image}_t), \quad P(x_i|x_j) \leftarrow \frac{\hat{n}(x_i, x_j)}{\sum_{x_i'} \hat{n}(x_i', x_j)}$$

Example cont'd

- We can also use the belief propagation algorithm to complete a partially observed image by evaluating $P(x_i|\text{partial image})$ for each unobserved leaf node