CS281A/Stat241A Lecture 20

Examples of Junction Tree Algorithm

Peter Bartlett

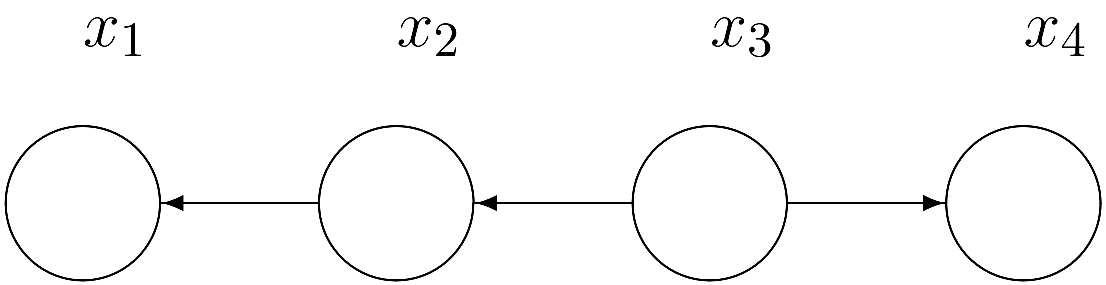
Announcements

- My office hours:
  Thursday Nov 5 (today), 1-2pm, in 723 Sutardja Dai Hall.
- Homework 5 due 5pm Monday, November 16.

Key ideas of this lecture

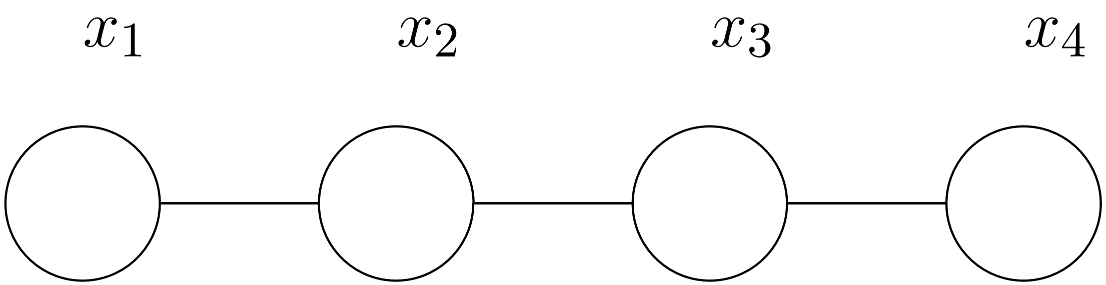
- Examples of the Junction Tree Algorithm.
  - Inference in a tree: Sum-product.
  - HMM
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to forward-backward.
  - Linear Dynamical Systems
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to Kalman filter/smoother.

Inference in a Tree: Sum-Product



$$p(x) = p(x_3)p(x_2|x_3)p(x_1|x_2)p(x_4|x_3)$$

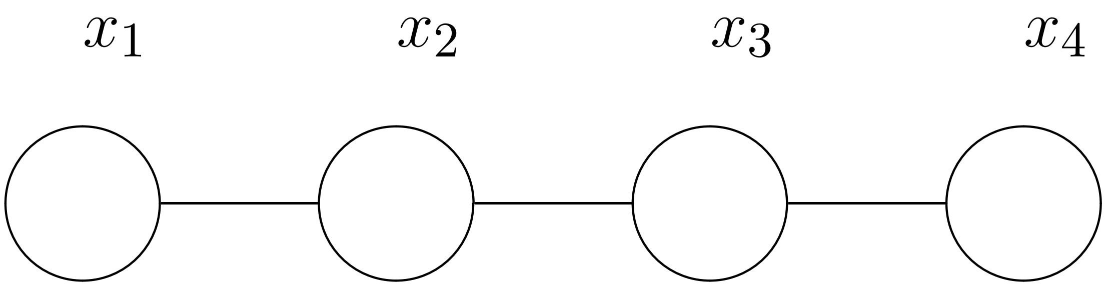
Inference in a Tree: Sum-Product



$$p(x) = \underbrace{p(x_3)p(x_2|x_3)}\, p(x_1|x_2)p(x_4|x_3)$$

$$= \psi_{23}(x_2, x_3)\psi_{12}(x_1, x_2)\psi_{34}(x_3, x_4).$$

Inference in a Tree: Sum-Product

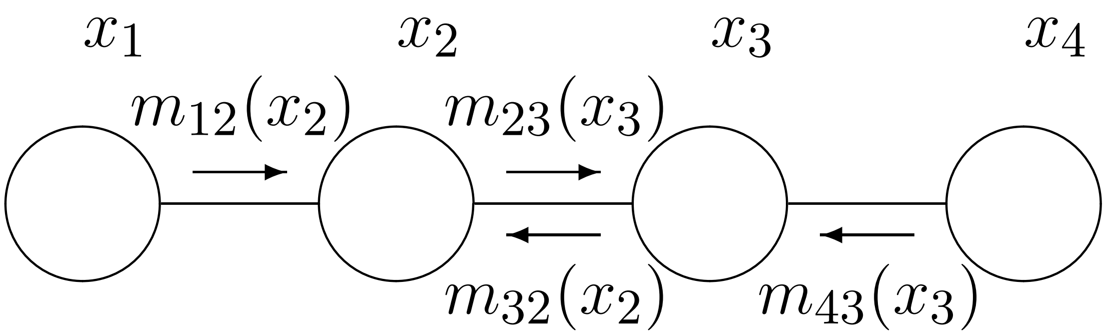


$$m_{j,i}(x_i) = \sum_{x_j} \left(\psi(x_i, x_j) \prod_{k \in N(j) \setminus \{i\}} m_{k,j}(x_j) \right),$$

$$p(x_i) = \prod_{j \in N(i)} m_{j,i}(x_i).$$

where $N(j) = \{k : \{k, j\} \in \mathcal{E}\}$.

Inference in a Tree: Sum-Product



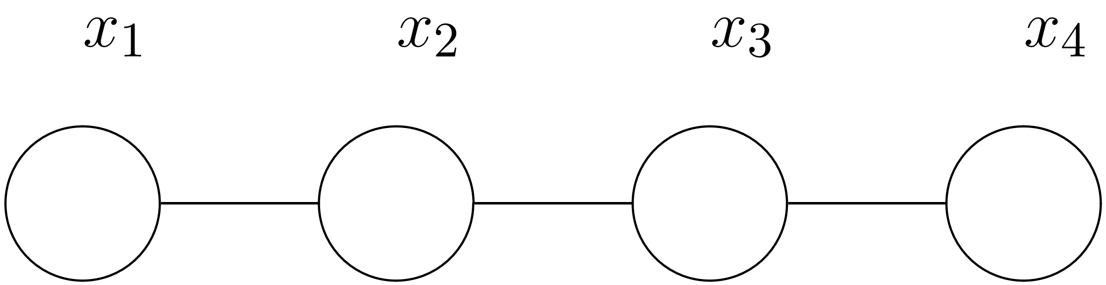
$$m_{12}(x_2) = \sum_{x_1} \psi_{12}(x_1, x_2)$$

$$m_{23}(x_3) = \sum_{x_2} \psi_{23}(x_2, x_3) m_{12}(x_2)$$

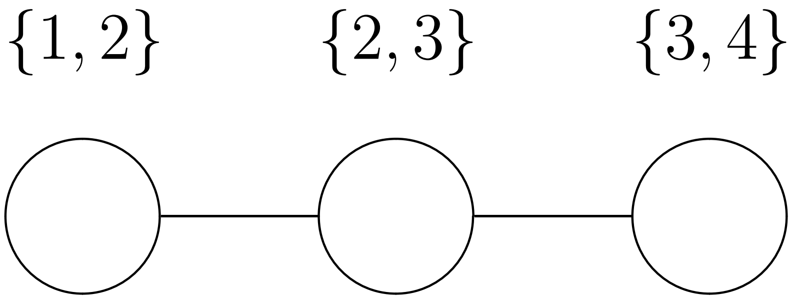
$$m_{43}(x_3) = \sum_{x_4} \psi_{34}(x_3, x_4)$$

$$m_{32}(x_2) = \sum_{x_3} \psi_{23}(x_2, x_3) m_{43}(x_3)$$

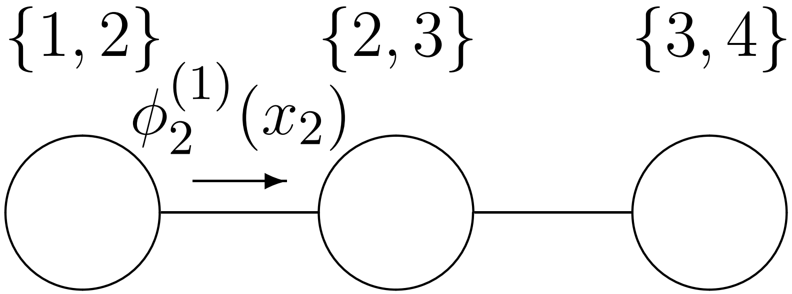
Inference in a Tree: Junction Tree



Junction Tree:



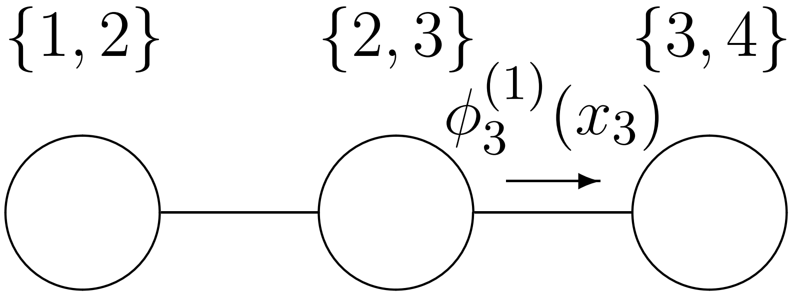
Inference in a Tree: Junction Tree



$$\phi_2^{(1)}(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \qquad = m_{12}(x_2).$$

$$\psi_{2,3}^{(1)} = \psi_{2,3}\phi_2^{(1)}.$$

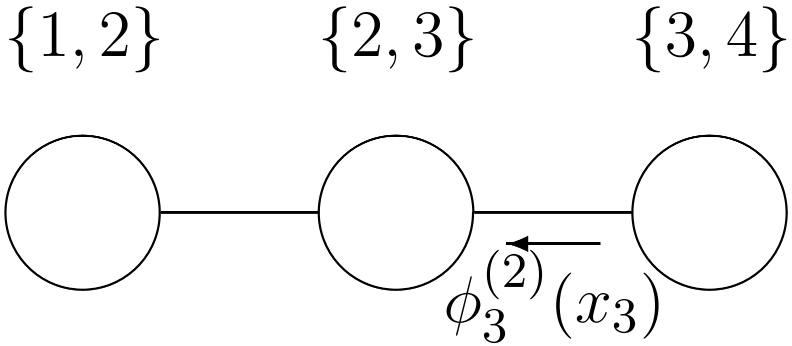
Inference in a Tree: Junction Tree



$$\phi_3^{(1)}(x_3) = \sum_{x_2} \psi_{2,3}(x_2, x_3) \qquad = m_{23}(x_3).$$

$$\psi_{3,4}^{(1)} = \psi_{3,4}\phi_3^{(1)}.$$

Inference in a Tree: Junction Tree

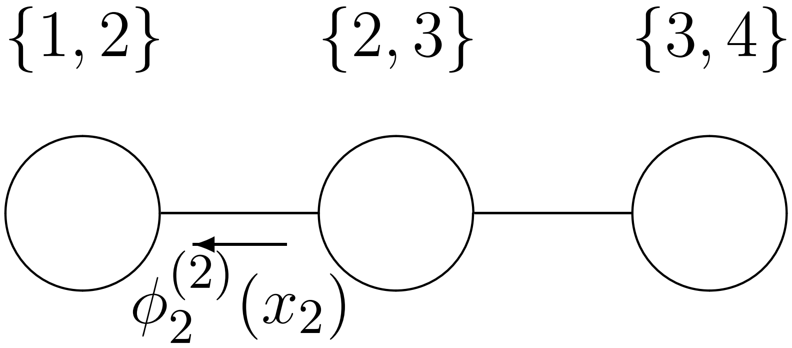


$$\phi_3^{(2)}(x_3) = \sum_{x_4} \psi_{3,4}^{(1)}(x_3, x_4)$$

$$= \phi_3^{(1)} \sum_{x_4} \psi_{3,4} \qquad = \phi_3^{(1)} m_{43}(x_3).$$

$$\psi_{2,3}^{(2)} = \psi_{2,3}^{(1)} \frac{\phi_3^{(2)}}{\phi_3^{(1)}}.$$

Inference in a Tree: Junction Tree

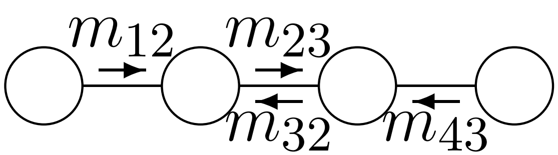


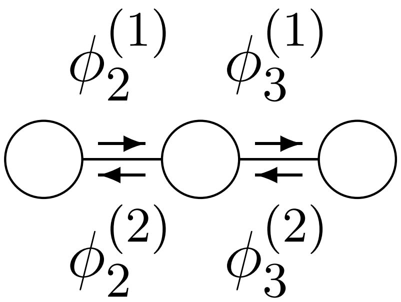
$$\phi_2^{(2)}(x_2) = \sum_{x_3} \psi_{2,3}^{(2)}(x_2, x_3)$$

$$= \phi_2^{(1)} \sum_{x_3} \psi_{2,3} \frac{\phi_3^{(2)}}{\phi_3^{(1)}}$$

$$= \phi_2^{(1)} \sum_{x_3} \psi_{2,3} m_{4,3} \qquad = \phi_2^{(1)} m_{32}.$$

Inference in a Tree: Junction Tree





$$m_{1,2} = \phi_2^{(1)},$$

$$m_{3,2} = \frac{\phi_2^{(2)}}{\phi_2^{(1)}}.$$

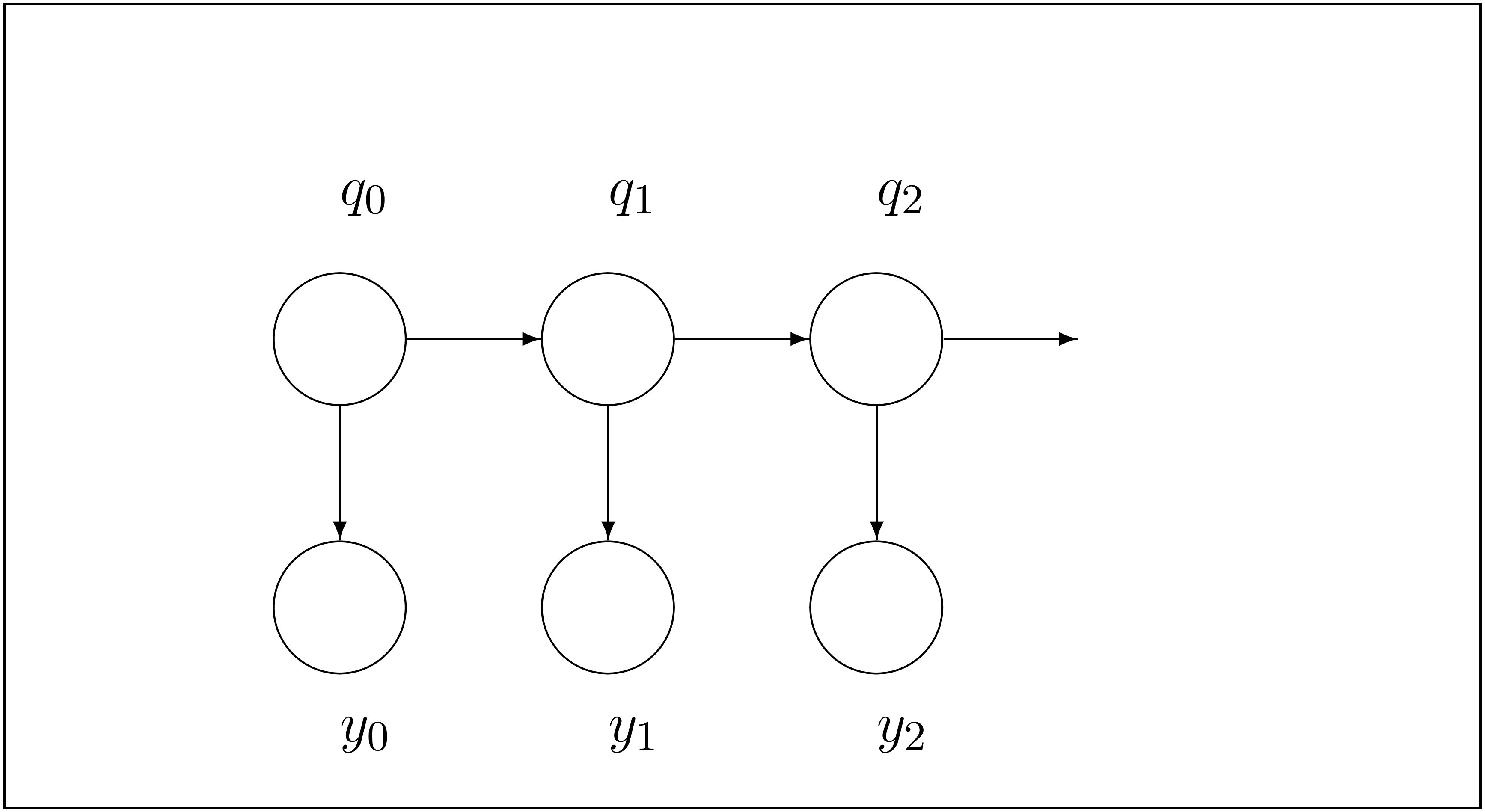
$$p(x_2) = m_{1,2}(x_2) m_{3,2}(x_2) \qquad = \phi_2^{(2)}(x_2).$$

Similarly for bigger trees: $m_{i,j}$ is a ratio $\phi_j^{(t+1)}/\phi_j^{(t)}$.

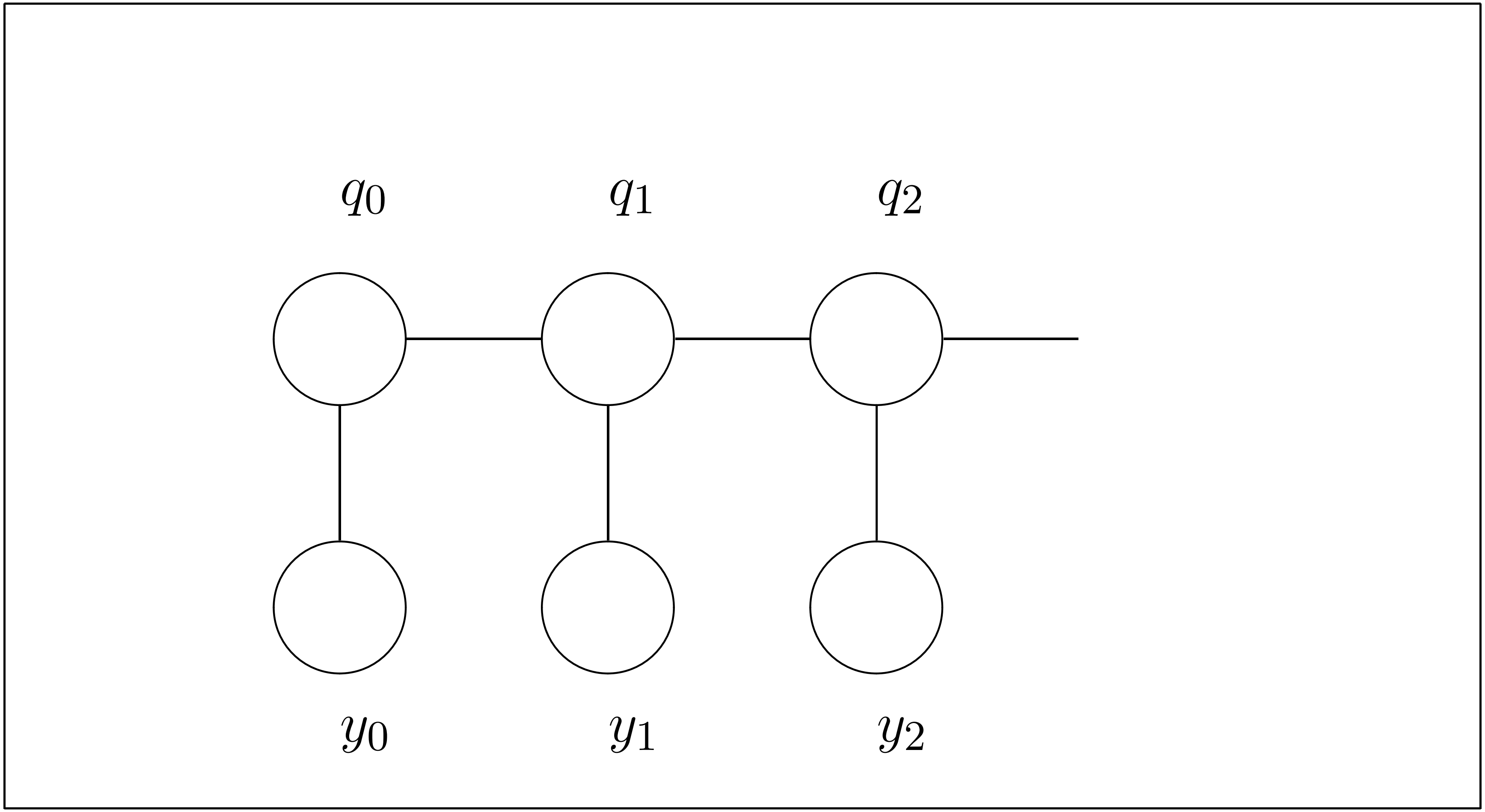
Key ideas of this lecture

- Examples of the Junction Tree Algorithm.
  - Inference in a tree: Sum-product.
  - HMM
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to forward-backward.
  - Linear Dynamical Systems
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to Kalman filter/smoother.

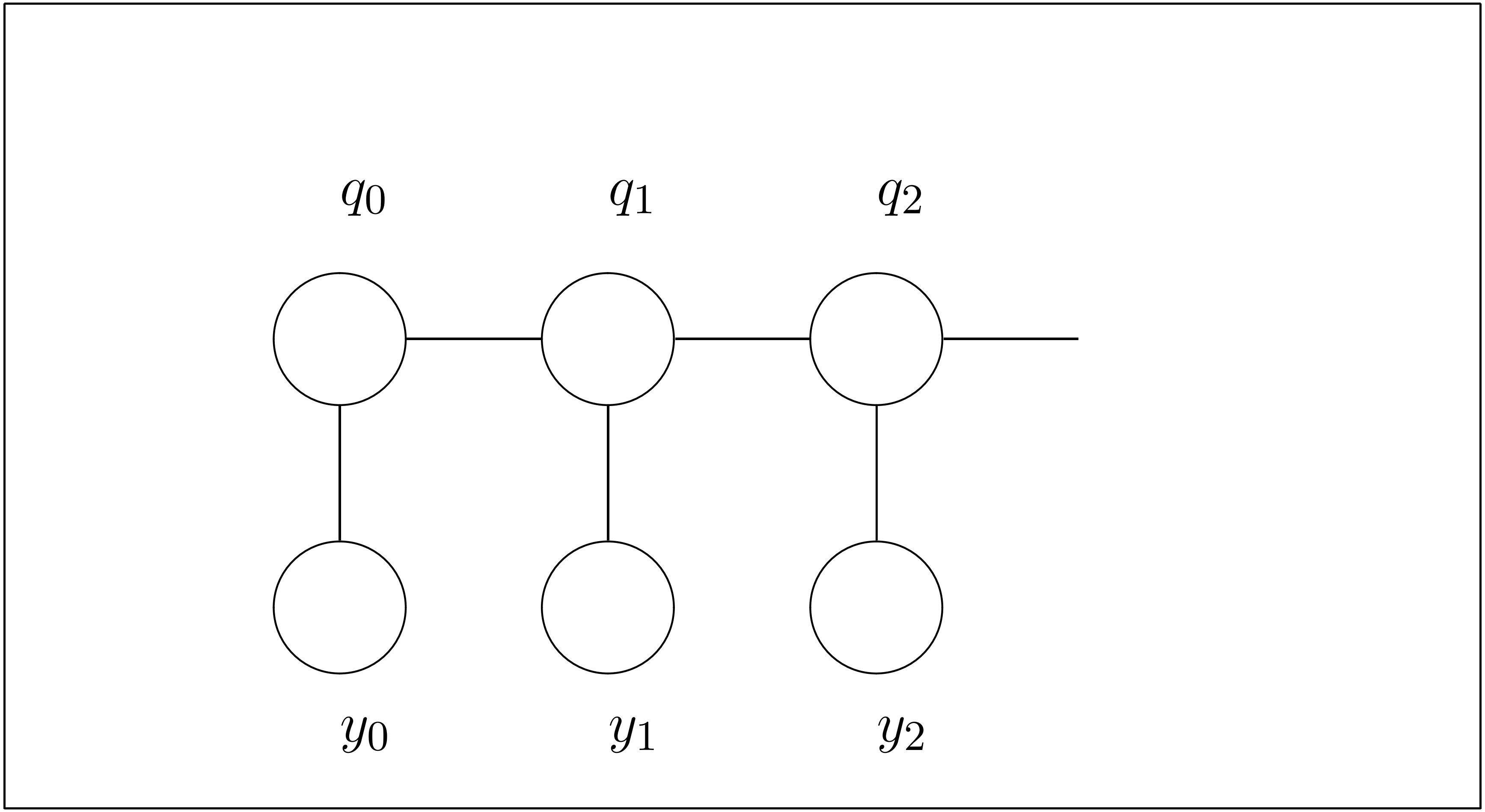
Hidden Markov Model



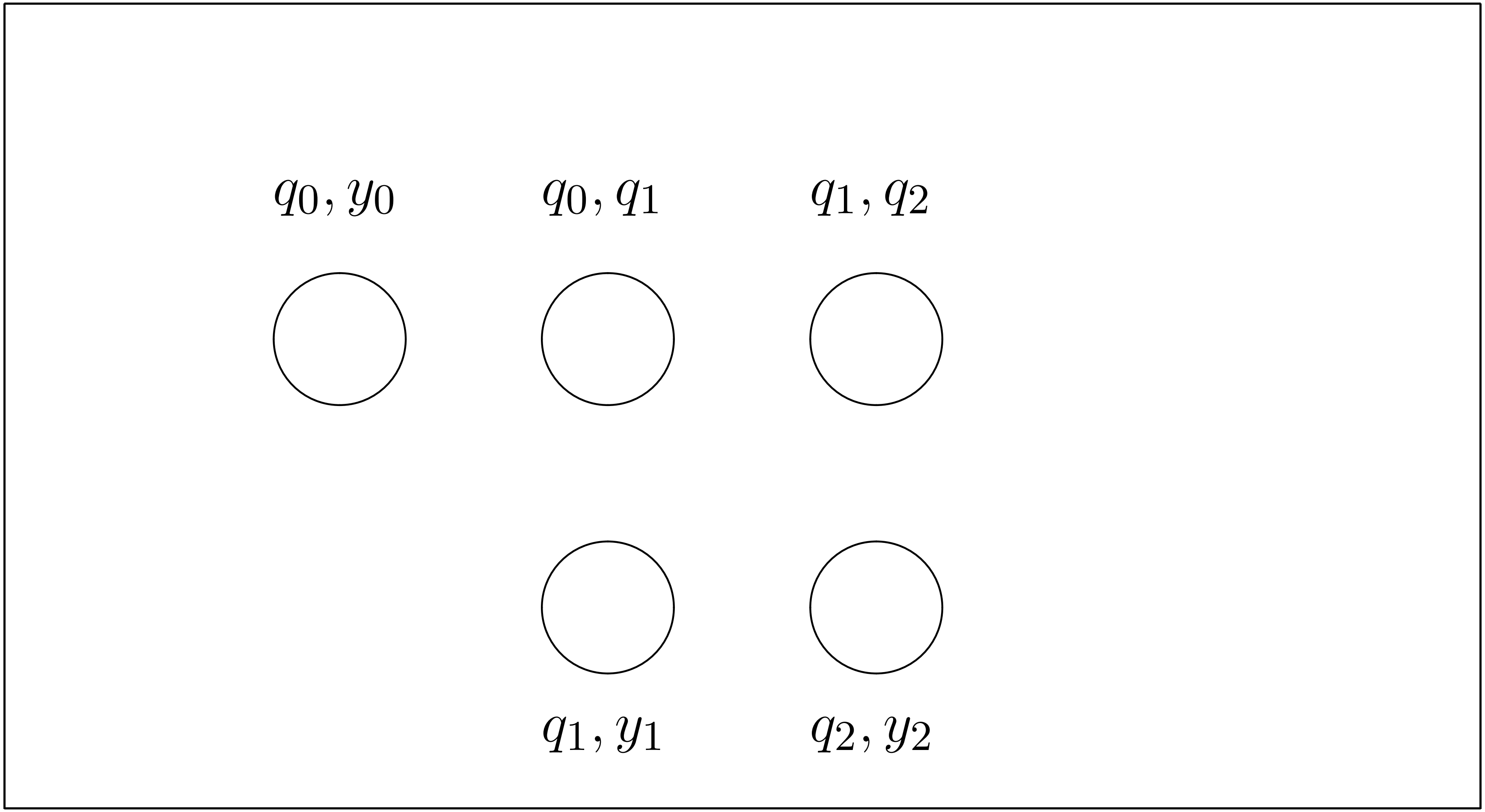
HMM: Moralize



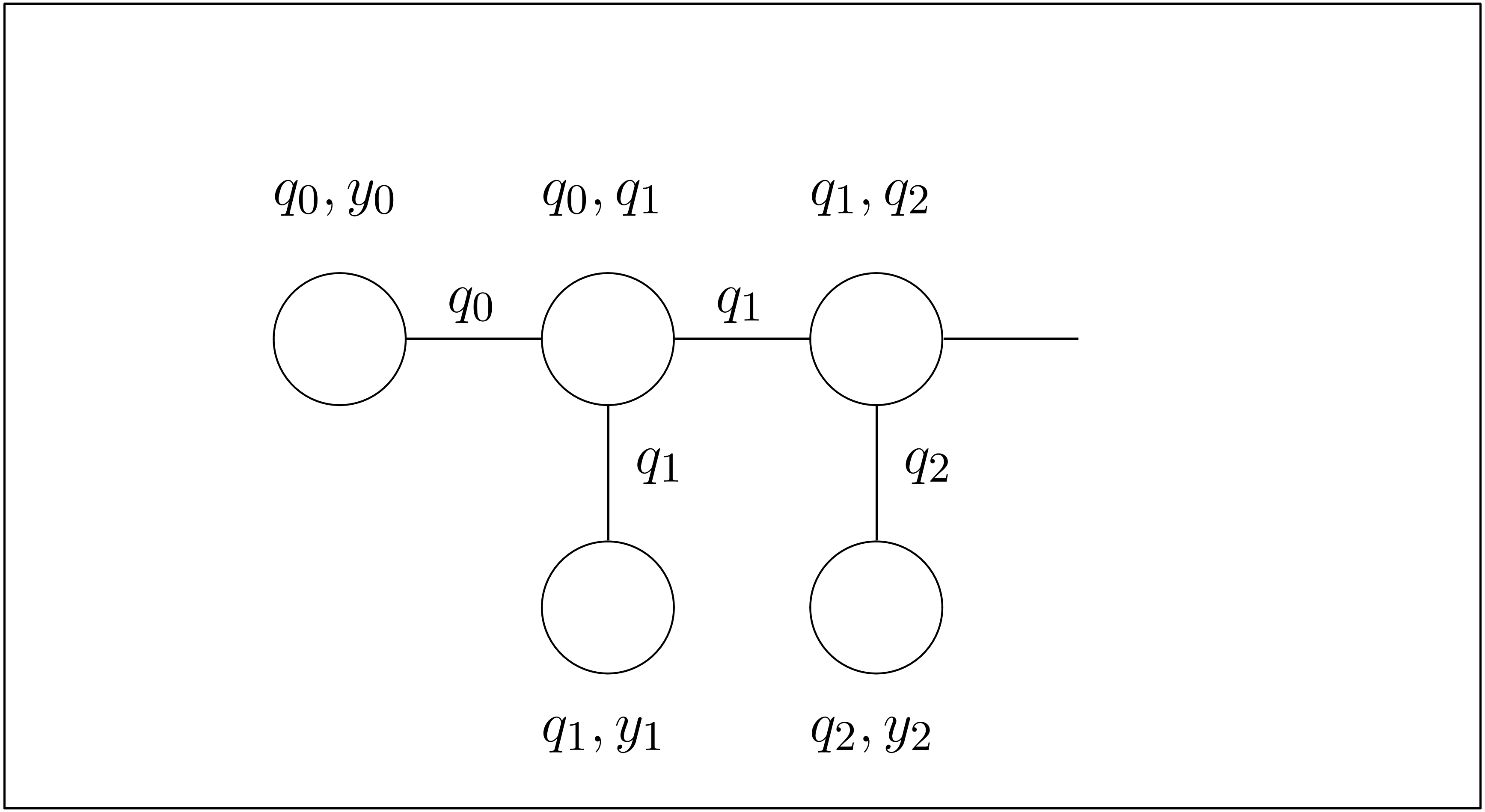
HMM: Triangulate



HMM: Clique Tree



HMM: Spanning Tree



- All other potential edges have weight zero (and lead to a cycle).

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree.
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

HMM: Potentials

Potentials:

$$p(q_0, y_0, q_1, y_1, q_2, y_2)$$
$$= \underbrace{p(q_0)p(y_0|q_0)} p(q_1|q_0)p(y_1|q_1)p(q_2|q_1)p(y_2|q_2)$$

$$\psi(q_0, y_0) = p(q_0)p(y_0|q_0)$$
$$\psi(q_t, q_{t+1}) = p(q_{t+1}|q_t)$$
$$\psi(q_t, y_t) = p(y_t|q_t) \qquad \text{(for } t \geq 1\text{)}$$
$$\phi(q_t) = 1.$$

Evidence: Observe $y_t = \bar{y}_t$, $t = 1, 2, \ldots$

Recall: Junction Tree Algorithm

- A message is passed from clique V to adjacent clique W (once V has received messages from all its other neighbors).
- The message corresponds to the updates:

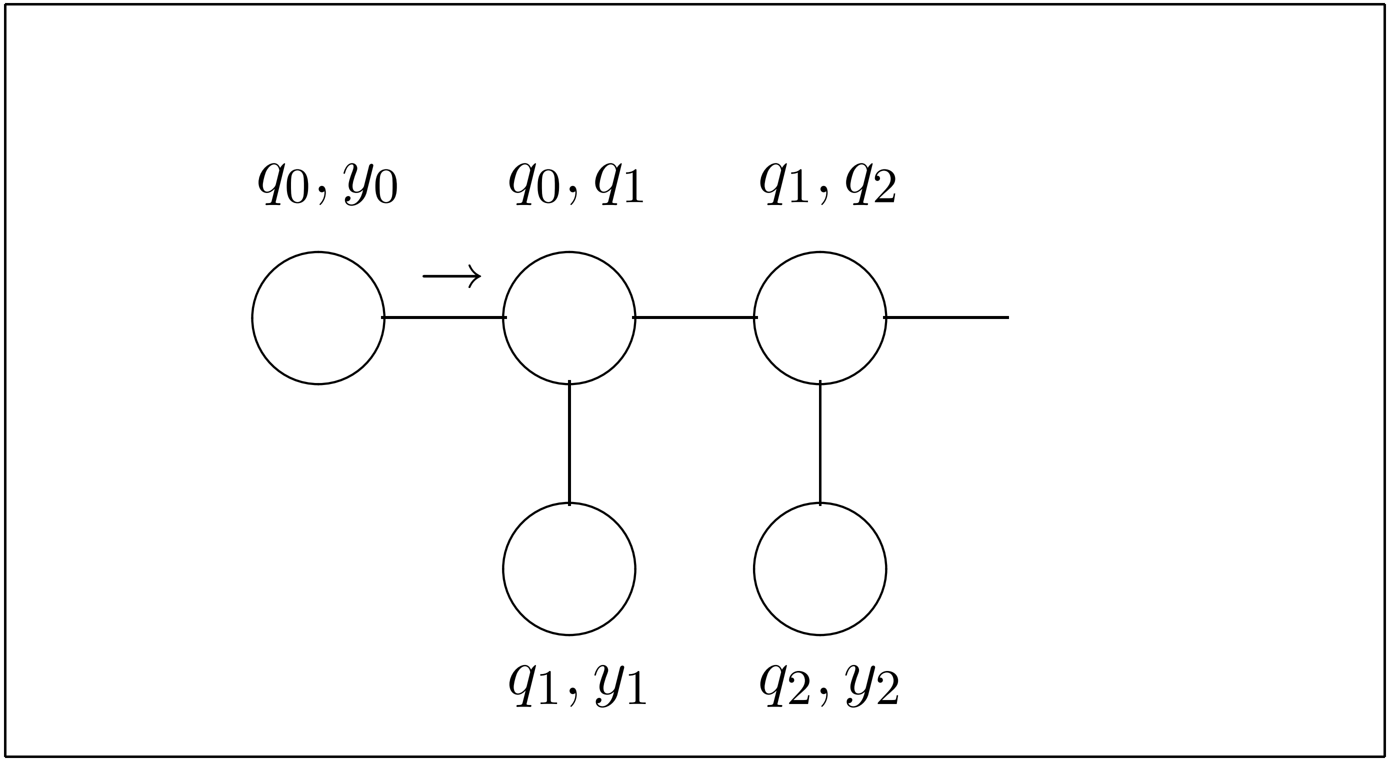
$$\phi_S^{(1)}(x_S) = \sum_{x_{V-S}} \psi_V(x_V),$$

$$\psi_W^{(1)}(x_W) = \psi_W(x_W) \frac{\phi_S^{(1)}(x_S)}{\phi_S(x_S)},$$

$$\psi_V^{(1)}(x_V) = \psi_V(x_V),$$

where $S = V \cap W$ is the separator.

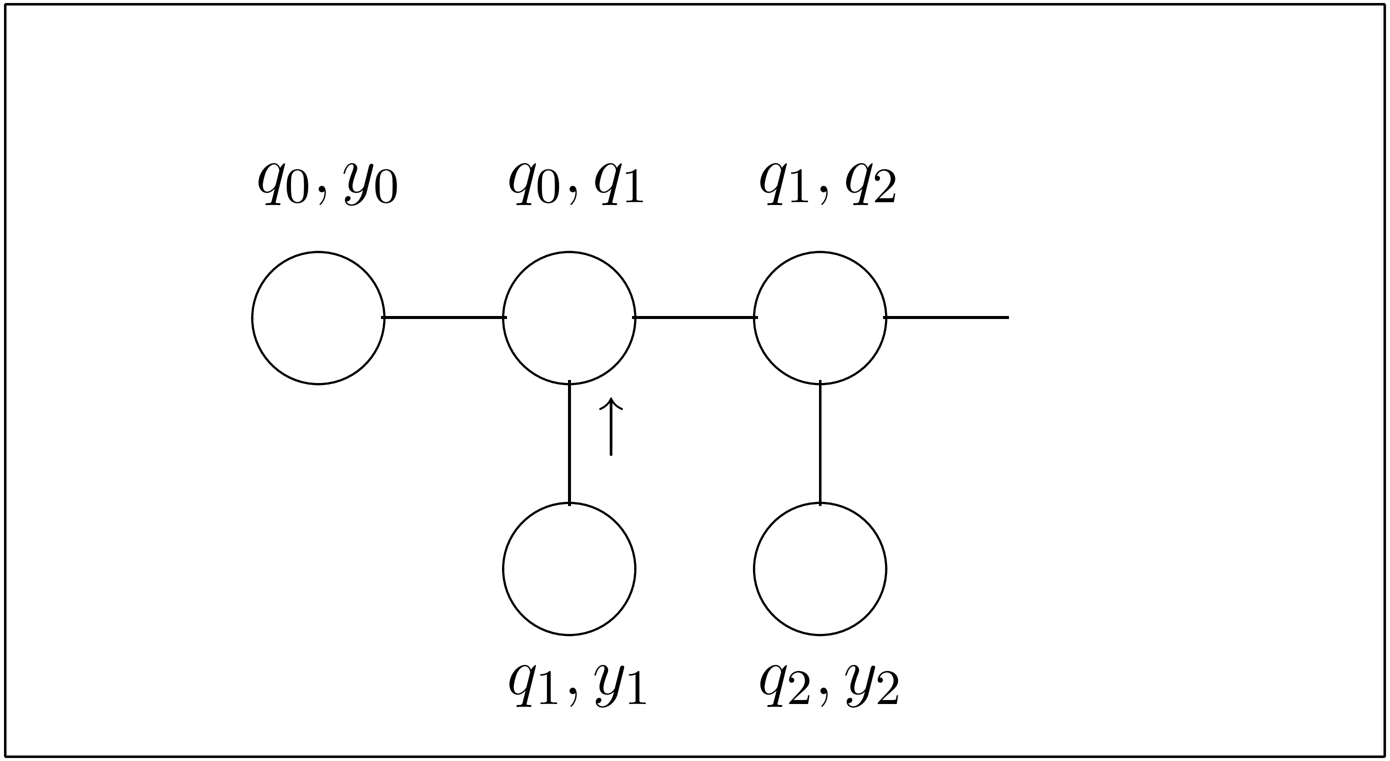
HMM: Propagate Probabilities



$$\phi_{\rightarrow}^{(1)}(q_0) := \pi_{q_0} p(\bar{y}_0 | q_0),$$

$$\psi^{(1)}(q_0, q_1) := a_{q_0,q_1} \pi_{q_0} p(\bar{y}_0 | q_0).$$

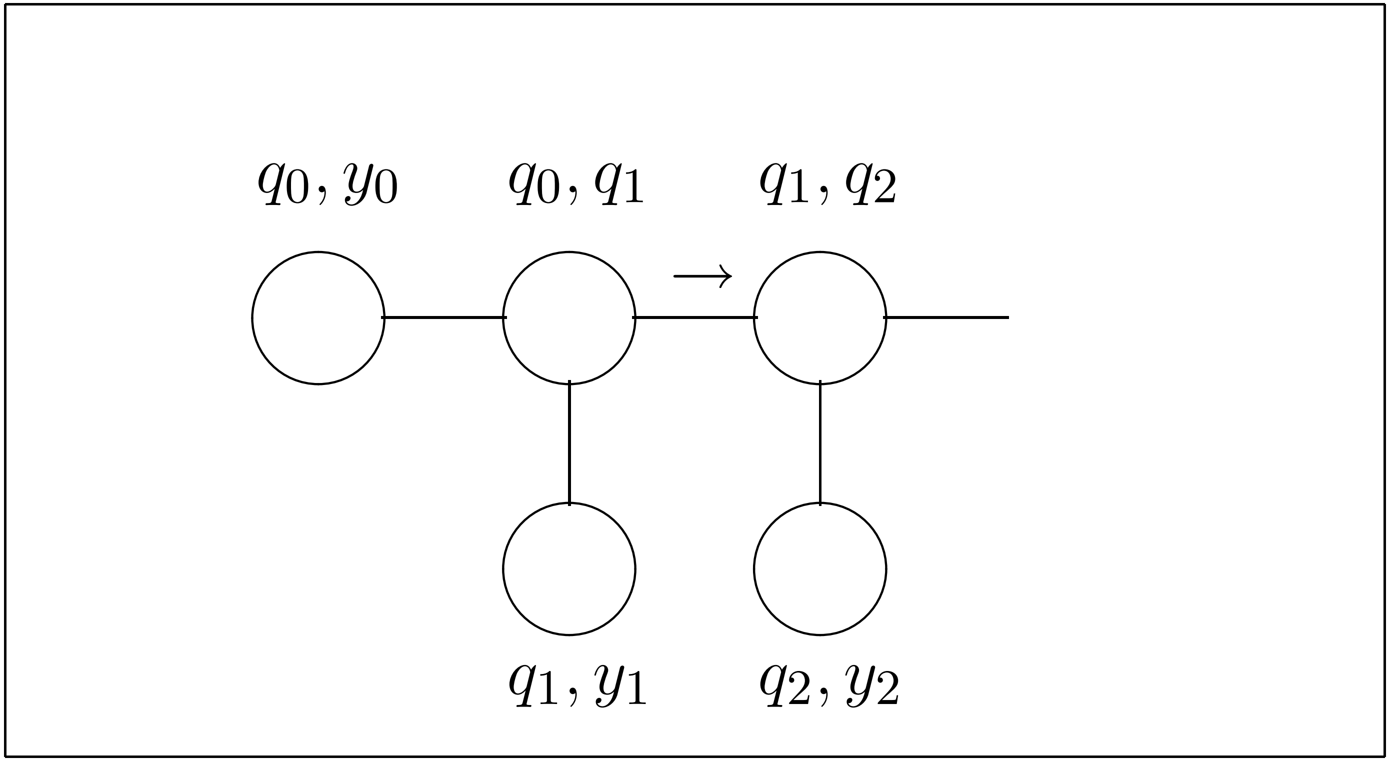
HMM: Propagate Probabilities



$$\phi_{\uparrow}^{(1)}(q_1) := p(\bar{y}_1 | q_1),$$

$$\psi^{(2)}(q_0, q_1) := a_{q_0, q_1} \pi_{q_0} p(\bar{y}_0 | q_0) p(\bar{y}_1 | q_1).$$

HMM: Propagate Probabilities



$$\phi^{(1)}_{\rightarrow}(q_{t+1}) := \sum_{q_t} \psi^{(2)}(q_t, q_{t+1})$$

$$= \sum_{q_t} a_{q_t, q_{t+1}} \phi^{(1)}_{\rightarrow}(q_t) p(\bar{y}_{t+1} | q_{t+1}).$$

HMM: Propagate Probabilities

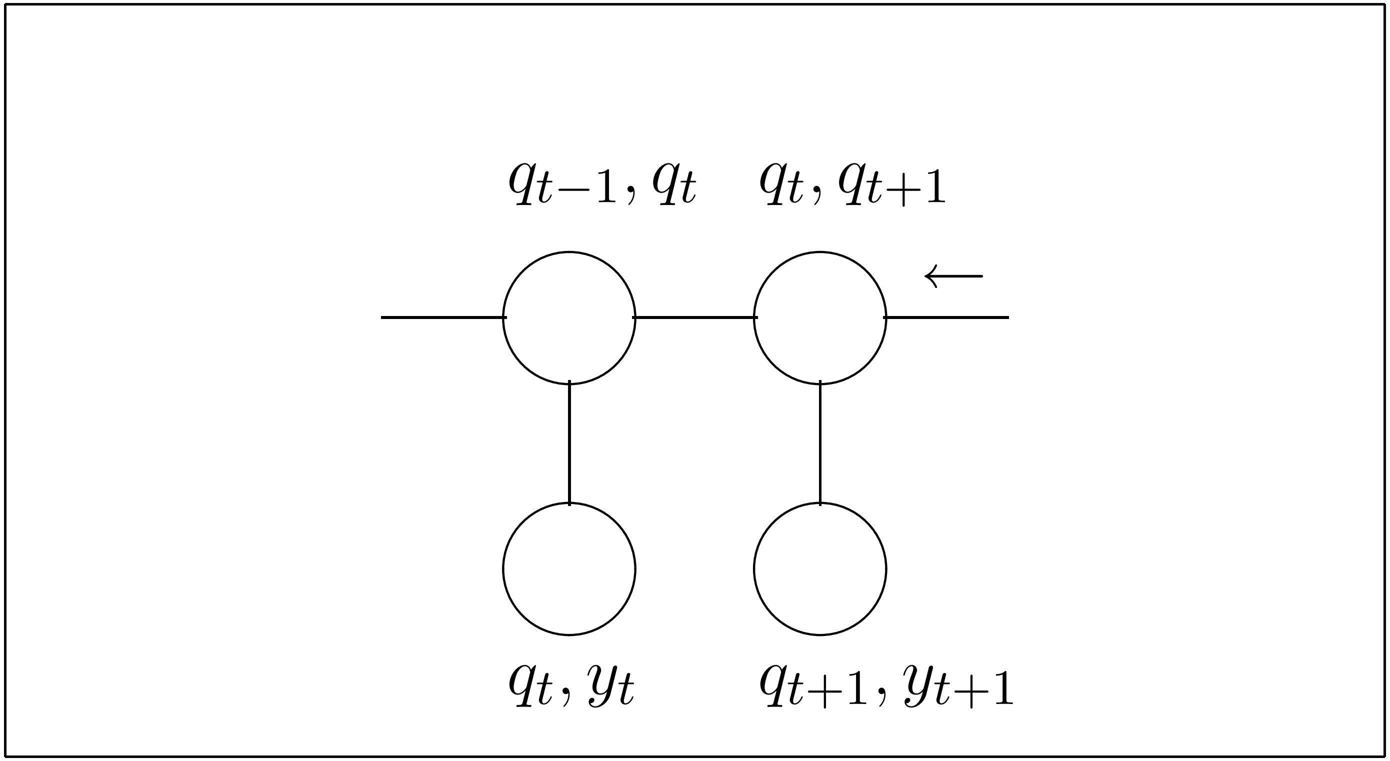
$$\phi_{\rightarrow}^{(1)}(q_{t+1}) = \sum_{q_t} a_{q_t,q_{t+1}} \phi_{\rightarrow}^{(1)}(q_t) p(\bar{y}_{t+1}|q_{t+1}).$$

- With $\alpha = \phi_{\rightarrow}$, this is the α-iteration of forward-backward:

$$\alpha(q_t) = p(\bar{y}_t|q_t) \sum_{q_{t-1}} a_{q_{t-1},q_t} \alpha(q_{t-1}),$$

$$\alpha(q_t) = p(q_t, y_1, \ldots, y_t).$$

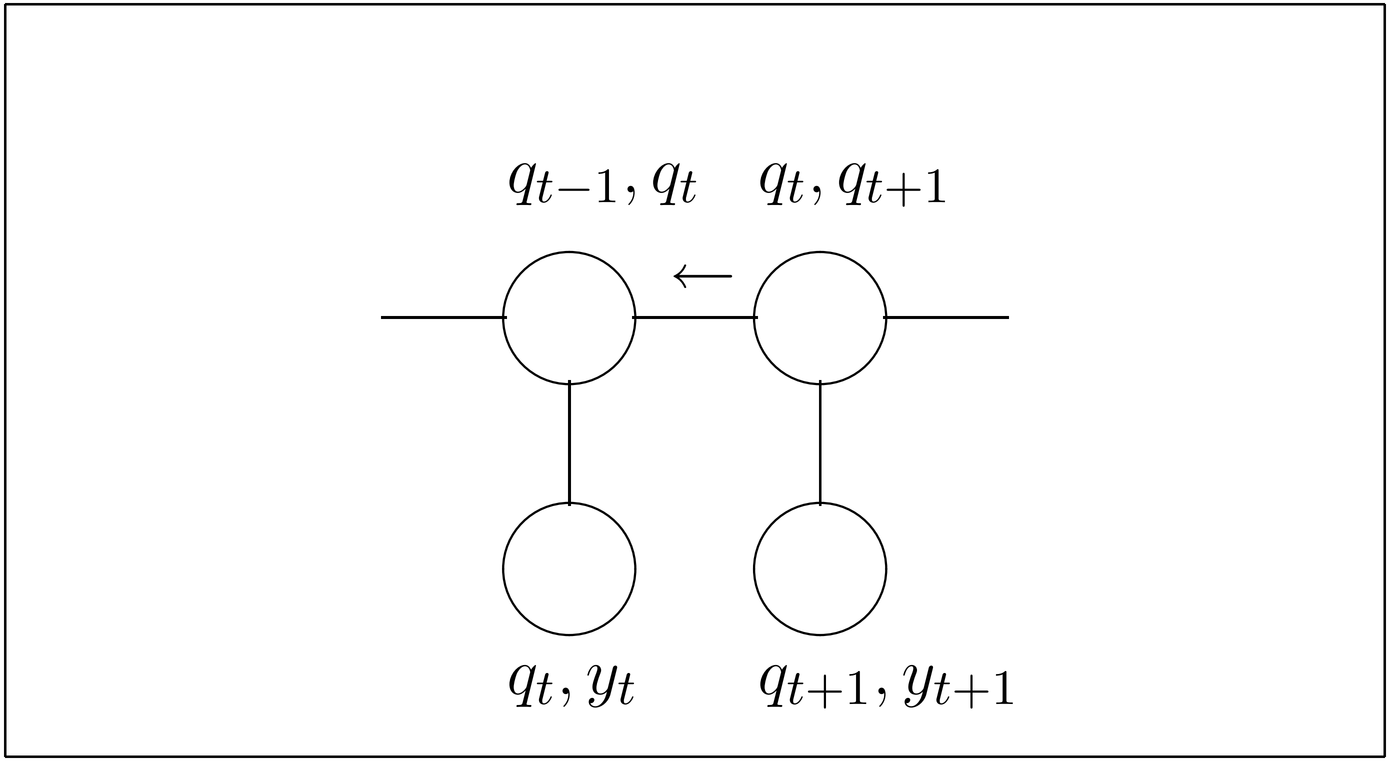
HMM: Propagate Backwards



$$\phi_{\rightarrow}^{(2)}(q_{t+1}) := \cdots$$

$$\psi^{(3)}(q_t, q_{t+1}) := \psi^{(2)}(q_t, q_{t+1}) \frac{\phi_{\rightarrow}^{(2)}(q_{t+1})}{\phi_{\rightarrow}^{(1)}(q_{t+1})}$$

HMM: Propagate Backwards



$$\phi^{(2)}_{\rightarrow}(q_t) := \sum_{q_{t+1}} \psi^{(3)}(q_t, q_{t+1})$$

HMM: Propagate Backwards

$$\phi_{\rightarrow}^{(2)}(q_t) := \sum_{q_{t+1}} \psi^{(3)}(q_t, q_{t+1})$$

$$= \sum_{q_{t+1}} \psi^{(2)}(q_t, q_{t+1}) \frac{\phi_{\rightarrow}^{(2)}(q_{t+1})}{\phi_{\rightarrow}^{(1)}(q_{t+1})}$$

$$= \sum_{q_{t+1}} \psi^{(2)}(q_t, q_{t+1}) \frac{\phi_{\rightarrow}^{(2)}(q_{t+1})}{\sum_{q_t} \psi^{(2)}(q_t, q_{t+1})}$$

$$= \sum_{q_{t+1}} \frac{a_{q_t,q_{t+1}} \phi_{\rightarrow}^{(1)}(q_t) p(\bar{y}_{t+1}|q_{t+1}) \phi_{\rightarrow}^{(2)}(q_{t+1})}{\sum_{q_t} a_{q_t,q_{t+1}} \phi_{\rightarrow}^{(1)}(q_t) p(\bar{y}_{t+1}|q_{t+1})}$$

$$= \sum_{q_{t+1}} \frac{a_{q_t,q_{t+1}} \phi_{\rightarrow}^{(1)}(q_t) \phi_{\rightarrow}^{(2)}(q_{t+1})}{\sum_{q_t} a_{q_t,q_{t+1}} \phi_{\rightarrow}^{(1)}(q_t)}.$$

HMM: Forward-Backward

$$\phi^{(2)}_{\rightarrow}(q_t) := \sum_{q_{t+1}} \frac{a_{q_t,q_{t+1}} \phi^{(1)}_{\rightarrow}(q_t) \phi^{(2)}_{\rightarrow}(q_{t+1})}{\sum_{q_t} a_{q_t,q_{t+1}} \phi^{(1)}_{\rightarrow}(q_t)}.$$

- With $\alpha = \phi^{(1)}_{\rightarrow}$ and $\gamma \propto \phi^{(2)}_{\rightarrow}$, this is the γ-iteration of forward-backward.

$$\gamma(q_t) = \sum_{q_{t+1}} \frac{a_{q_t,q_{t+1}} \alpha(q_t) \gamma(q_{t+1})}{\sum_{q_t} a_{q_t,q_{t+1}} \alpha(q_t)},$$

$$\gamma(q_t) = p(q_t | y_0, \ldots, y_T).$$

- γ and $\phi^{(2)}_{\rightarrow}$ are identical up to a scaling factor.

HMM: Forward-Backward

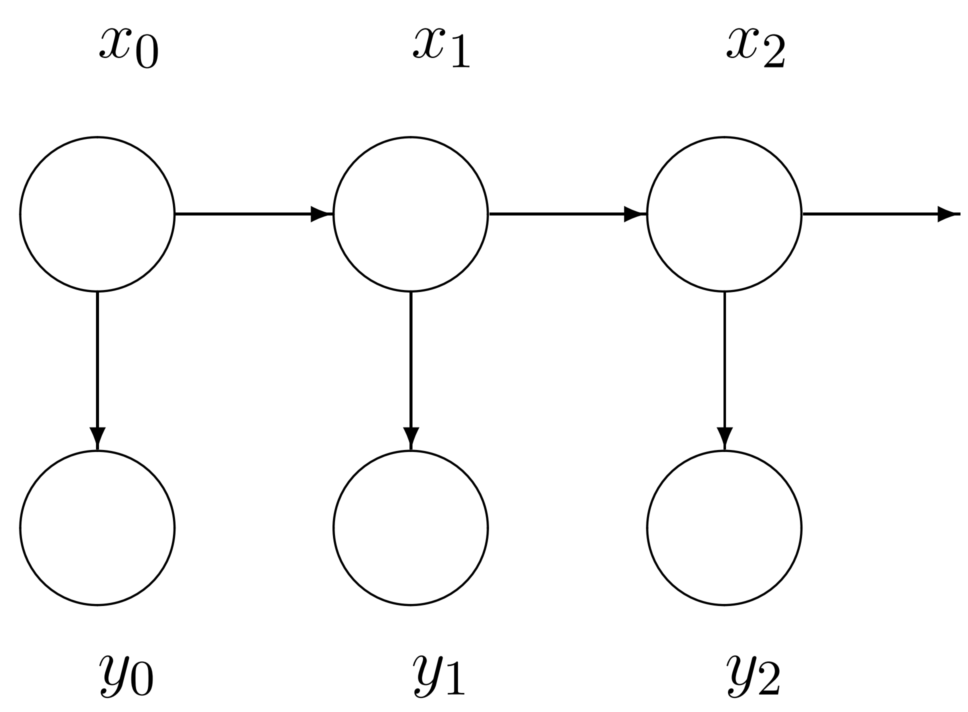
- Updating the $\phi_{\rightarrow}$s when passing messages forwards corresponds to the α-recursion.
- Updating the $\phi_{\rightarrow}$s when passing messages backwards corresponds to the γ-recursion.
- The updated ψs, after passing messages backwards, correspond to the ξs in forward-backward:

$$\xi(q_t, q_{t+1}) = p(q_t, q_{t+1} | y_1, \ldots, y_T).$$

Key ideas of this lecture

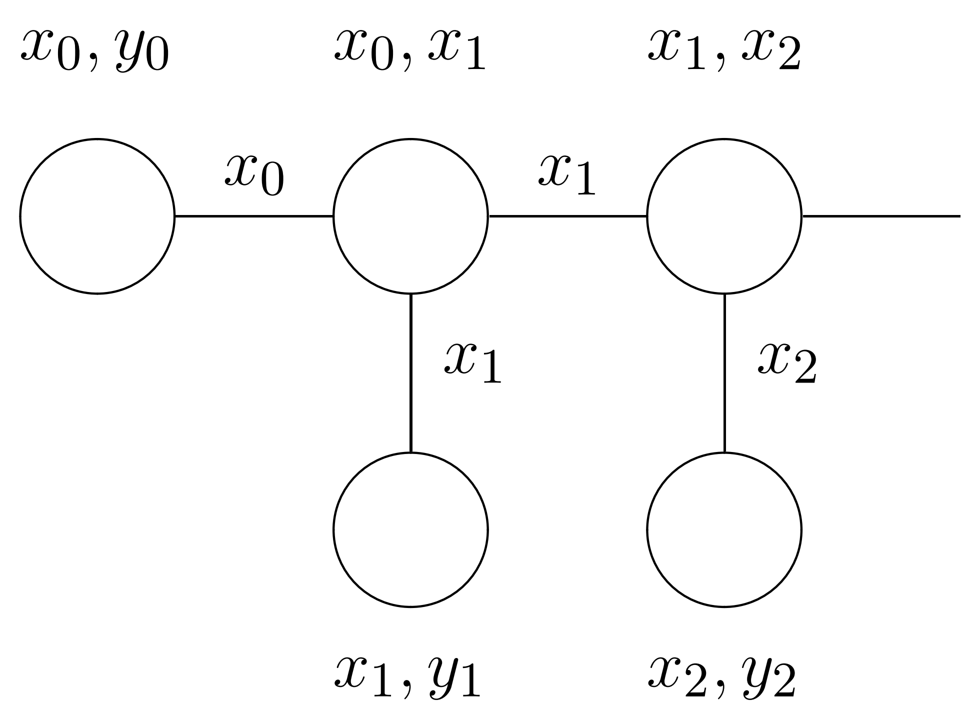
- Examples of the Junction Tree Algorithm.
  - HMM
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to forward-backward.
  - Linear Dynamical Systems
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to Kalman filter/smoother.

Linear Dynamical System



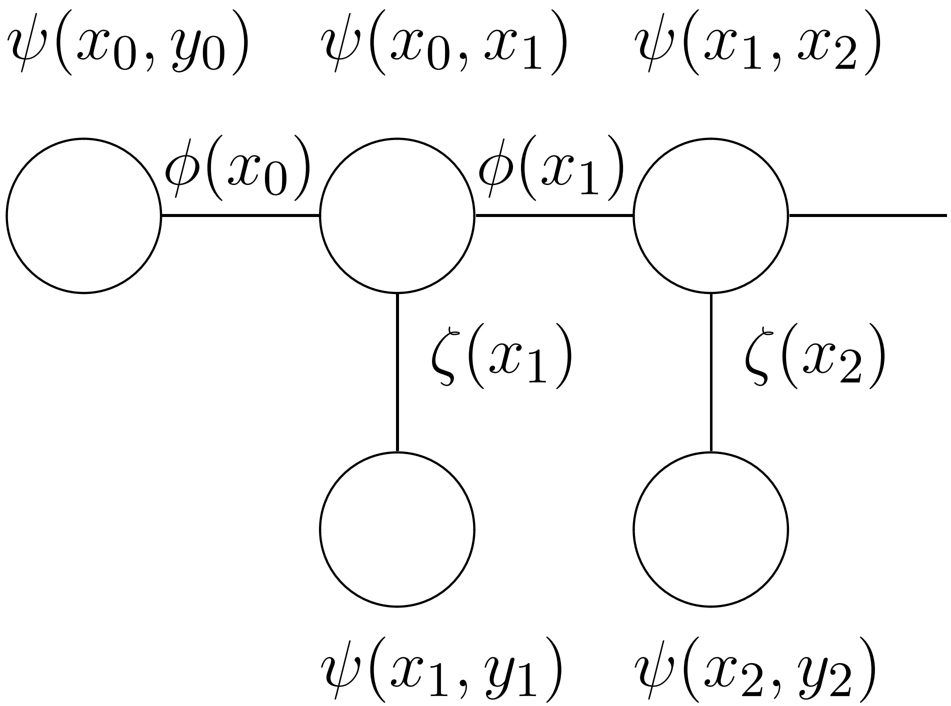
Linear Dynamical System: Junction Tree

Moralize, triangulate, form junction tree:



Linear Dynamical System: Potentials

Define potentials:



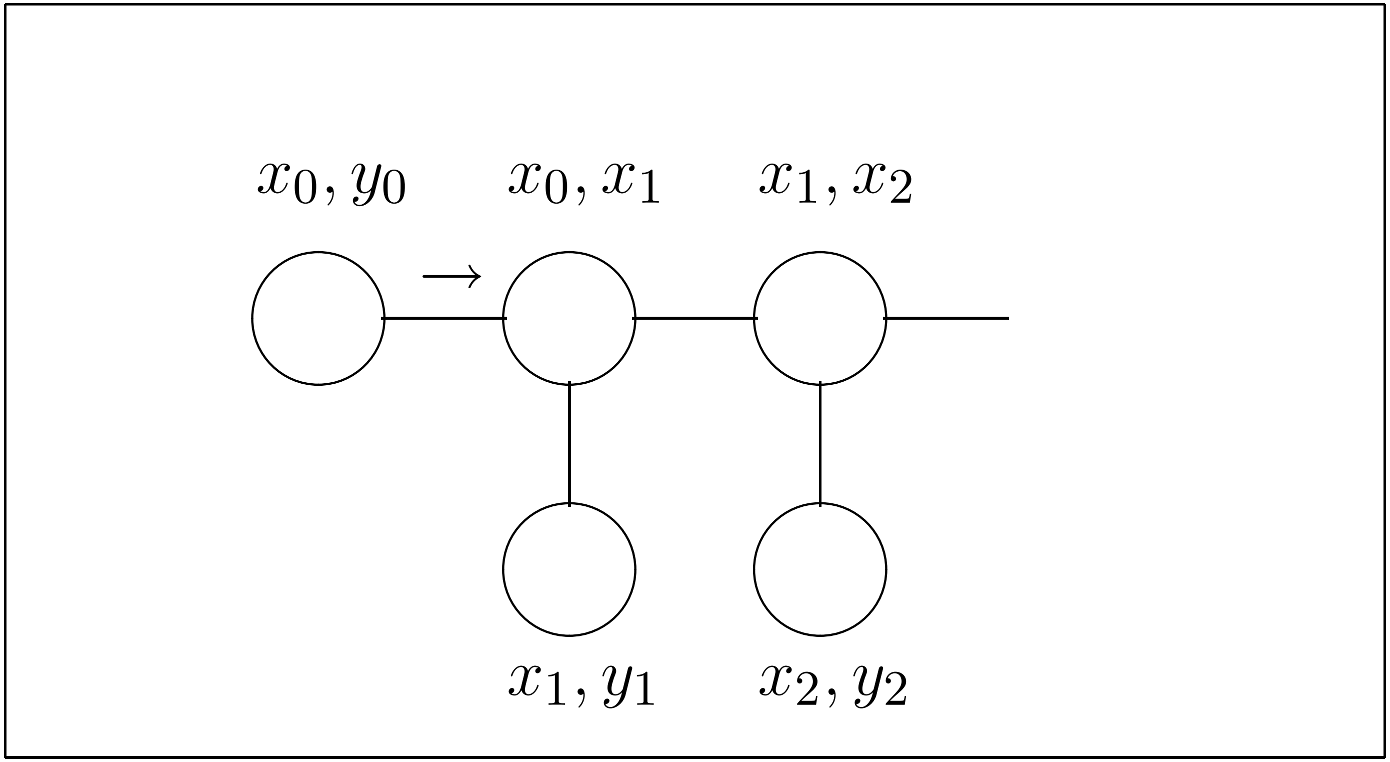
Linear Dynamical System: Potentials

$$
\begin{aligned}
\phi(x_i) &= \zeta(x_i) = 1, \\
\psi(x_0, y_0) &= p(x_0)p(y_0|x_0), \\
\psi(x_t, y_t) &= p(y_t|x_t) \qquad\qquad 1 \leq t \leq T, \\
\psi(x_{t-1}, x_t) &= p(x_t|x_{t-1}).
\end{aligned}
$$

- The last three expressions are Gaussians, represented using the parameters.
- When messages are passed and potentials updated, the parameters of the Gaussians are updated.

Linear Dynamical System: Messages

Propagate messages towards $\{x_{T-1}, x_T\}$**:**



$$\phi^{(1)}(x_0) = p(x_0, \bar{y}_0) \propto p(x_0|\bar{y}_0).$$

Linear Dynamical System: Messages

$$\psi(x_0, y_0) = p(x_0)p(y_0|x_0),$$
$$= \mathcal{N}(x_0; 0, P_0)\mathcal{N}(y_0; Cx_0, R)$$

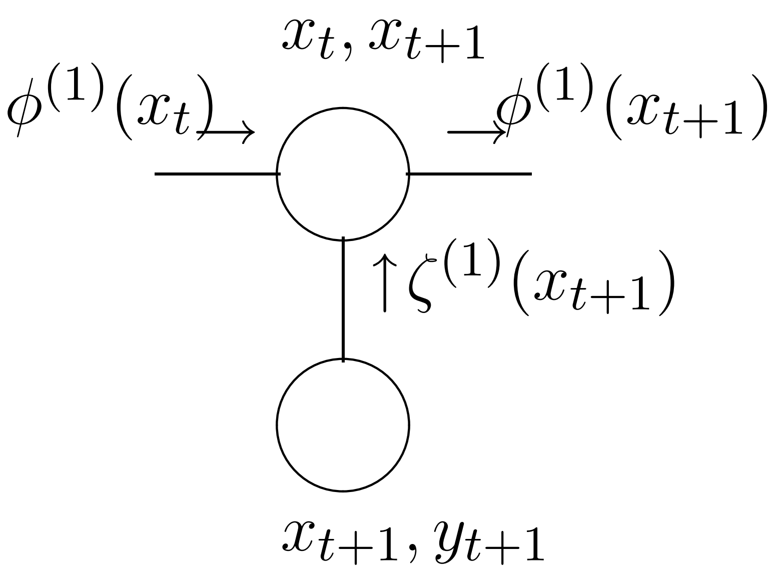
$$\mu = 0, \qquad \Sigma = \begin{pmatrix} P_0 & P_0C' \\ CP_0 & CP_0C' + R \end{pmatrix}$$

Marginalizing (integrating over y_0) corresponds to conditioning on evidence:

$$\phi^{(1)}(x_0) \propto p(x_0|\bar{y}_0) :$$
$$\hat{x}_{0|0} = P_0C'(CP_0C' + R)^{-1}\bar{y}_0$$
$$P_{0|0} = P_0 - P_0C'(CP_0C' + R)^{-1}CP_0.$$

Linear Dynamical System: Messages

Propagate messages towards $\{x_{T-1}, x_T\}$**:**



Suppose: $\phi^{(1)}(x_t) \propto p(x_t | \bar{y}_0, \ldots, \bar{y}_t)$.

Show: $\phi^{(1)}(x_{t+1}) \propto p(x_{t+1} | \bar{y}_0, \ldots, \bar{y}_{t+1})$.

Messages: $\psi^{(1)}$

$$\begin{aligned}\psi^{(1)}(x_t, x_{t+1}) &= \psi(x_t, x_{t+1})\phi^{(1)}(x_t) \\ &\propto p(x_{t+1}|x_t)p(x_t|\bar{y}_0, \ldots, \bar{y}_t) \\ &= p(x_t, x_{t+1}|\bar{y}_0, \ldots, \bar{y}_t).\end{aligned}$$

Notice that, if we marginalized out x_t, this would give

$$p(x_{t+1}|\bar{y}_0, \ldots, \bar{y}_t),$$

that is, the ψ update incorporates a *time update*.

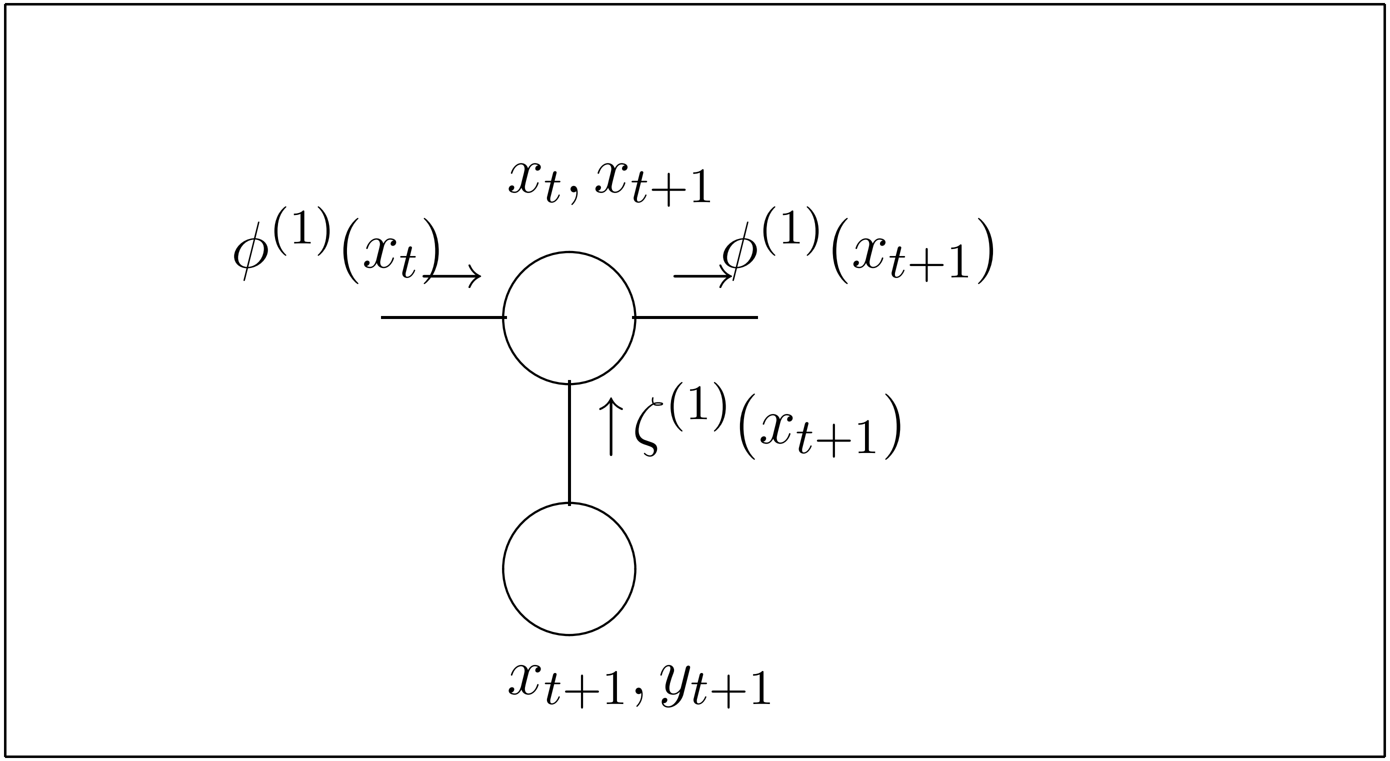
Messages: $\psi^{(1)}$

$$\psi(x_t, x_{t+1}) = p(x_{t+1}|x_t)$$
$$\propto \exp\left(-\frac{1}{2}(x_{t+1} - Ax_t)' Q^{-1} (x_{t+1} - Ax_t)\right)$$

$$\phi^{(1)}(x_t) = p(x_t|\bar{y}_0, \ldots, \bar{y}_t)$$
$$\propto \exp\left(-\frac{1}{2}\left(x_t - \hat{x}_{t|t}\right)' P_{t|t}^{-1} \left(x_t - \hat{x}_{t|t}\right)\right).$$

And so the product $\psi^{(1)}(x_t, x_{t+1}) = \psi(x_t, x_{t+1})\phi^{(1)}(x_t)$ is exponential in a quadratic (a sum of quadratics) in x_t and x_{t+1}.

Linear Dynamical System: Messages



Messages: $\zeta^{(1)}$, $\psi^{(2)}$, ϕ^1

$$\zeta^{(1)}(x_{t+1}) \propto p(\bar{y}_{t+1}|x_{t+1})$$

$$\begin{aligned}
\psi^{(2)}(x_t, x_{t+1}) &= \psi^{(1)}(x_t, x_{t+1})\zeta^{(1)}(x_{t+1}) \\
&\propto p(x_t, x_{t+1}, \bar{y}_0, \ldots, \bar{y}_t)p(\bar{y}_{t+1}|x_{t+1}) \\
&= p(x_t, x_{t+1}, \bar{y}_0, \ldots, \bar{y}_t)p(\bar{y}_{t+1}|x_t, x_{t+1}, \bar{y}_0, \ldots, \bar{y}_t) \\
&= p(x_t, x_{t+1}, \bar{y}_0, \ldots, \bar{y}_{t+1}) \\
&\propto p(x_t, x_{t+1}|\bar{y}_0, \ldots, \bar{y}_{t+1}).
\end{aligned}$$

$$\begin{aligned}
\phi^{(1)}(x_{t+1}) &= \int \psi^{(2)}(x_t, x_{t+1})dx_t \\
&\propto p(x_{t+1}|\bar{y}_0, \ldots, \bar{y}_{t+1}).
\end{aligned}$$

Linear Dynamical System: Messages

- We have: $\phi^{(1)}(x_t) \propto p(x_t|\bar{y}_0, \ldots, \bar{y}_t)$.
- Passing messages 'forward' in the junction tree corresponds to the Kalman filter.
- Messages are passed explicitly as parameters of the Gaussians:
  - $\psi(x_t, x_{t+1}), \psi(x_t, y_t)$ are Gaussian.
  - Conditioning on $\bar{y}_t$ $(\zeta^{(1)})$ gives a Gaussian.
  - Multiplying Gaussians (ψ) gives a Gaussian.
  - Marginalizing a Gaussian (ϕ) gives a Gaussian.

Linear Dynamical System

- Parameterization affects computation.
- Consider the multiplication of two Gaussians:

$$\exp\left(-\frac{1}{2}\left((x-\mu_1)'\Sigma_1^{-1}(x-\mu_1)+(x-\mu_2)'\Sigma_2^{-1}(x-\mu_2)\right)\right)$$
$$=\exp\left(-\frac{1}{2}\left(x'\left(\Sigma_1^{-1}+\Sigma_2^{-1}\right)x \right.\right.$$
$$\left.\left. -2\left(\Sigma_1^{-1}\mu_1+\Sigma_2^{-1}\mu_2\right)'x+\text{constants}\right)\right).$$

- Here, the natural parameterization is convenient:

$$\Lambda=\Lambda_1+\Lambda_2 \qquad \xi=\xi_1+\xi_2,$$
$$\text{where } \Lambda=\Sigma^{-1} \qquad \xi=\Sigma^{-1}\mu.$$

Linear Dynamical System

Choices:

- Parameterization:
  - Kalman filter
  - Information filter
- Junction tree, order of passing messages.

Key ideas of this lecture

- Examples of the Junction Tree Algorithm.
  - Inference in a tree: Sum-product.
  - HMM
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to forward-backward.
  - Linear Dynamical Systems
    - Construct junction tree,
    - Propagate probabilities,
    - Corresponds to Kalman filter/smoother.