Approximate inference and structural learning of MTE networks

S. Moral, V. Romero, R. Rumí, A. Salmerón

Departamento de Estadística y Matemática Aplicada

Universidad de Almería

Problems with continuous variables

- Methods for Bayesian networks usually require point-wise specified distributions.

Problems with continuous variables

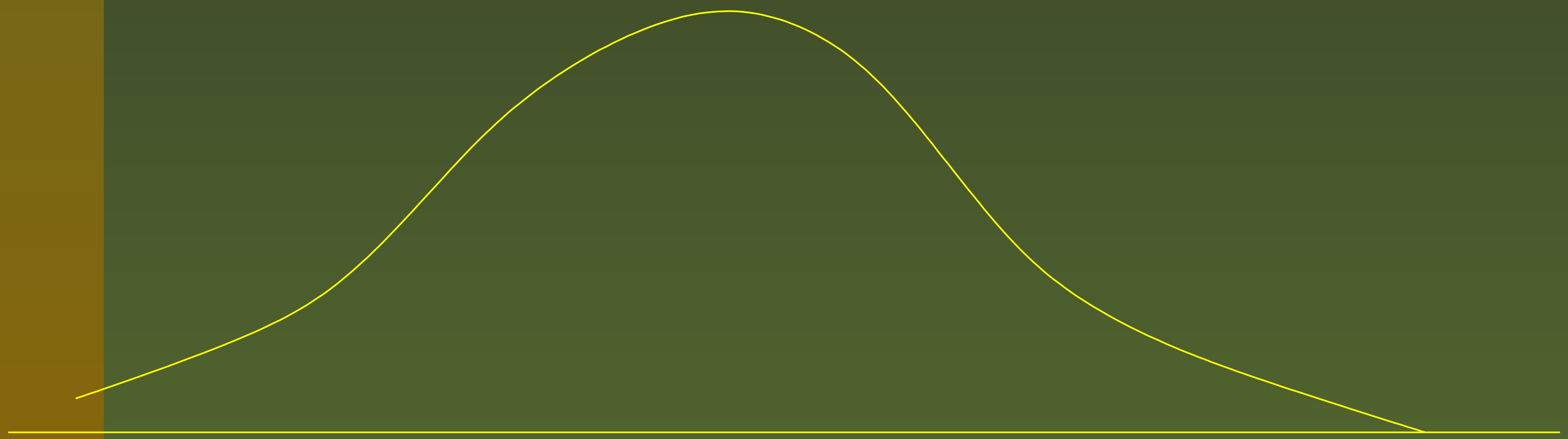
- Methods for Bayesian networks usually require point-wise specified distributions.
- If not, a general data structure for density functions should be found

Problems with continuous variables

- Methods for Bayesian networks usually require point-wise specified distributions.
- If not, a general data structure for density functions should be found
- Computation difficulties may arise.

Discretisation

$$f : \Omega \to \mathbb{R} \quad \text{density function.}$$

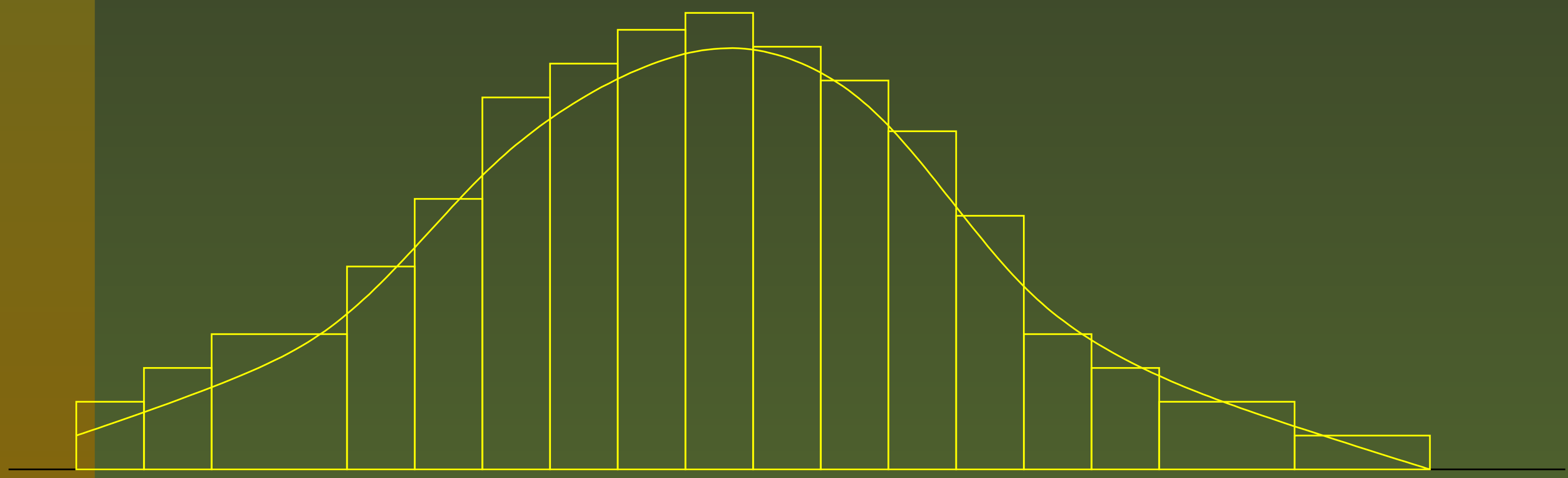


Discretisation

$$f : \Omega \to I\!R \quad \text{density function.}$$

$$\downarrow$$

$$f_D : \Omega \to I\!R \quad \text{histogram.}$$



Discretisation

- Universal.

Discretisation

- Universal.
- Allows for known techniques.

Discretisation

- Universal.
- Allows for known techniques.
- It is an approximation.

Conditional Gaussian distributions

Conditional Gaussian distributions

- Inference and learning depply studied.

Conditional Gaussian distributions

- Inference and learning depply studied.
- Restrictions on the network structure.

MTE potentials

- $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$: n-dimensional mixed random vector.

MTE potentials

- $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$: n-dimensional mixed random vector.
- $\phi : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$.

MTE potentials

- $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$: n-dimensional mixed random vector.
- $\phi : \Omega_{\mathbf{X}} \mapsto I\!R_0^+$.

i.

$$\phi(\mathbf{x}) = \phi(\mathbf{y}, \mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{d} b_i^{(j)} y_j + \sum_{k=1}^{c} b_i^{(d+k)} z_k \right\}$$

MTE potentials

- $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$: n-dimensional mixed random vector.
- $\phi : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$.

i.
$$\phi(\mathbf{x}) = \phi(\mathbf{y}, \mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{d} b_i^{(j)} y_j + \sum_{k=1}^{c} b_i^{(d+k)} z_k \right\}$$

ii. There is a partition $\Omega_1, \ldots, \Omega_k$ of $\Omega_{\mathbf{X}}$ such that $\phi(\mathbf{x}) = \phi_i(\mathbf{x}) \quad \text{if} \ \ \mathbf{x} \in \Omega_i$, where ϕ_i can be written as in i.

MTE potentials: example

$$\phi(x, z_1, z_2) = \begin{cases} 2 + e^{3z_1+z_2} & \text{if } \ 0 < z_1 \leq 1, \ 0 < z_2 < 2, \ x = 0 \\ 1 + e^{z_1+z_2} & \text{if } \ 0 < z_1 \leq 1, \ 2 \leq z_2 < 3, \ x = 0 \\ \frac{1}{4} + e^{2z_1+z_2} & \text{if } \ 1 < z_1 < 2, \ 0 < z_2 < 2, \ x = 0 \\ \frac{1}{2} + 5e^{z_1+2z_2} & \text{if } \ 1 < z_1 < 2, \ 2 \leq z_2 < 3, \ x = 0 \\ e^{z_1} + 2e^{z_2} & \text{if } \ 0 < z_1 \leq 1, \ x = 1 \\ 1 + e^{3z_1+z_2} & \text{if } \ 1 < z_1 < 2, \ x = 1 \end{cases}$$

MTE potentials: operations

- Restriction: $\mathbf{X}' \subseteq \mathbf{X}$ with $(\mathbf{x}')$ fixed.
  $\phi^{R(\mathbf{X}'=\mathbf{x}')}(\mathbf{w}) = \phi(\mathbf{w}, \mathbf{x}')$

MTE potentials: operations

- Restriction: $\mathbf{X}' \subseteq \mathbf{X}$ with $(\mathbf{x}')$ fixed.
  $\phi^{R(\mathbf{X}'=\mathbf{x}')}(\mathbf{w}) = \phi(\mathbf{w}, \mathbf{x}')$
- Marginalisation: $\mathbf{X}' = (\mathbf{Y}', \mathbf{Z}') \subseteq \mathbf{X}$.

$$\phi^{\downarrow \mathbf{X}'}(\mathbf{y}', \mathbf{z}') = \sum_{\mathbf{y} \in \Omega_{\mathbf{Y} \setminus \mathbf{Y}'}} \left(\int_{\Omega_{\mathbf{Z}''}} \phi(\mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}') d\mathbf{z}'' \right)$$

MTE potentials: operations

- Restriction: $\mathbf{X}' \subseteq \mathbf{X}$ with $(\mathbf{x}')$ fixed.
  $\phi^{R(\mathbf{X}'=\mathbf{x}')}(\mathbf{w}) = \phi(\mathbf{w}, \mathbf{x}')$
- Marginalisation: $\mathbf{X}' = (\mathbf{Y}', \mathbf{Z}') \subseteq \mathbf{X}$.

$$\phi^{\downarrow \mathbf{X}'}(\mathbf{y}', \mathbf{z}') = \sum_{\mathbf{y} \in \Omega_{\mathbf{Y} \setminus \mathbf{Y}'}} \left(\int_{\Omega_{\mathbf{Z}''}} \phi(\mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}') d\mathbf{z}'' \right)$$

- Combination:
  $\phi(\mathbf{x}) = \phi_1(\mathbf{x}^{\downarrow \Omega_{\mathbf{X_1}}}) \cdot \phi_2(\mathbf{x}^{\downarrow \Omega_{\mathbf{X_2}}})$ for all $\mathbf{x} \in \Omega_{\mathbf{X}}$.

MTE potentials: operations

- Restriction: $\mathbf{X}' \subseteq \mathbf{X}$ with $(\mathbf{x}')$ fixed.
  $\phi^{R(\mathbf{X}'=\mathbf{x}')}(\mathbf{w}) = \phi(\mathbf{w}, \mathbf{x}')$
- Marginalisation: $\mathbf{X}' = (\mathbf{Y}', \mathbf{Z}') \subseteq \mathbf{X}$.

$$\phi^{\downarrow \mathbf{X}'}(\mathbf{y}', \mathbf{z}') = \sum_{\mathbf{y} \in \Omega_{\mathbf{Y} \setminus \mathbf{Y}'}} \left(\int_{\Omega_{\mathbf{Z}''}} \phi(\mathbf{y}, \mathbf{y}', \mathbf{z}, \mathbf{z}') d\mathbf{z}'' \right)$$

- Combination:
  $\phi(\mathbf{x}) = \phi_1(\mathbf{x}^{\downarrow \Omega_{\mathbf{X_1}}}) \cdot \phi_2(\mathbf{x}^{\downarrow \Omega_{\mathbf{X_2}}})$ for all $\mathbf{x} \in \Omega_{\mathbf{X}}$.
- The class of MTE potentials is closed for the three operations.

MTE densities

- f is an MTE density if:

MTE densities

- f is an MTE density if:

  I. f is an MTE potential.

MTE densities

- f is an MTE density if:
  I. f is an MTE potential.
  II. $\displaystyle\sum_{\mathbf{y}\in\Omega_{\mathbf{Y}}} \int_{\Omega_{\mathbf{Z}}} f(\mathbf{y},\mathbf{z})d\mathbf{z} = 1$.

MTE densities

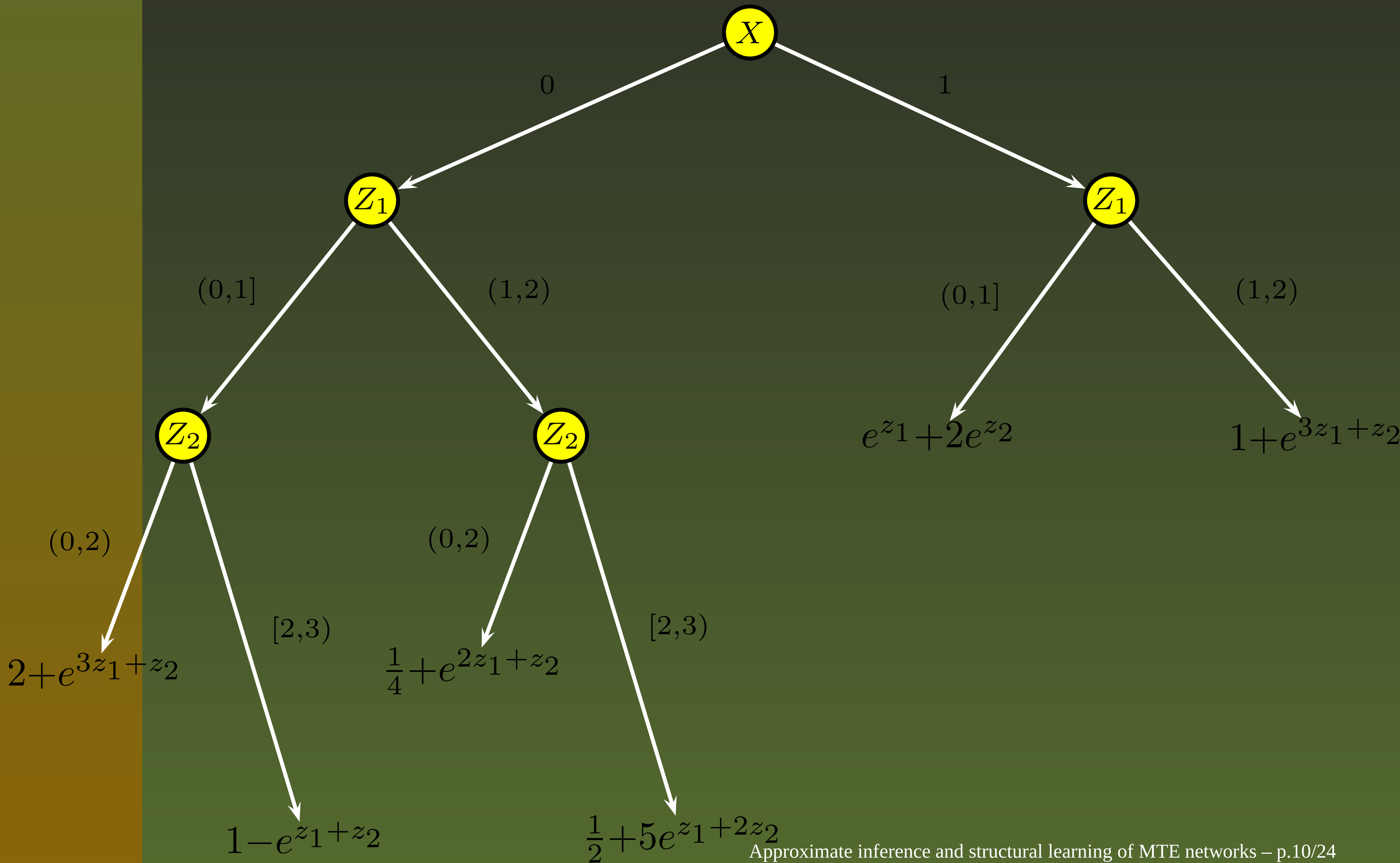
- f is an MTE density if:

  I. f is an MTE potential.

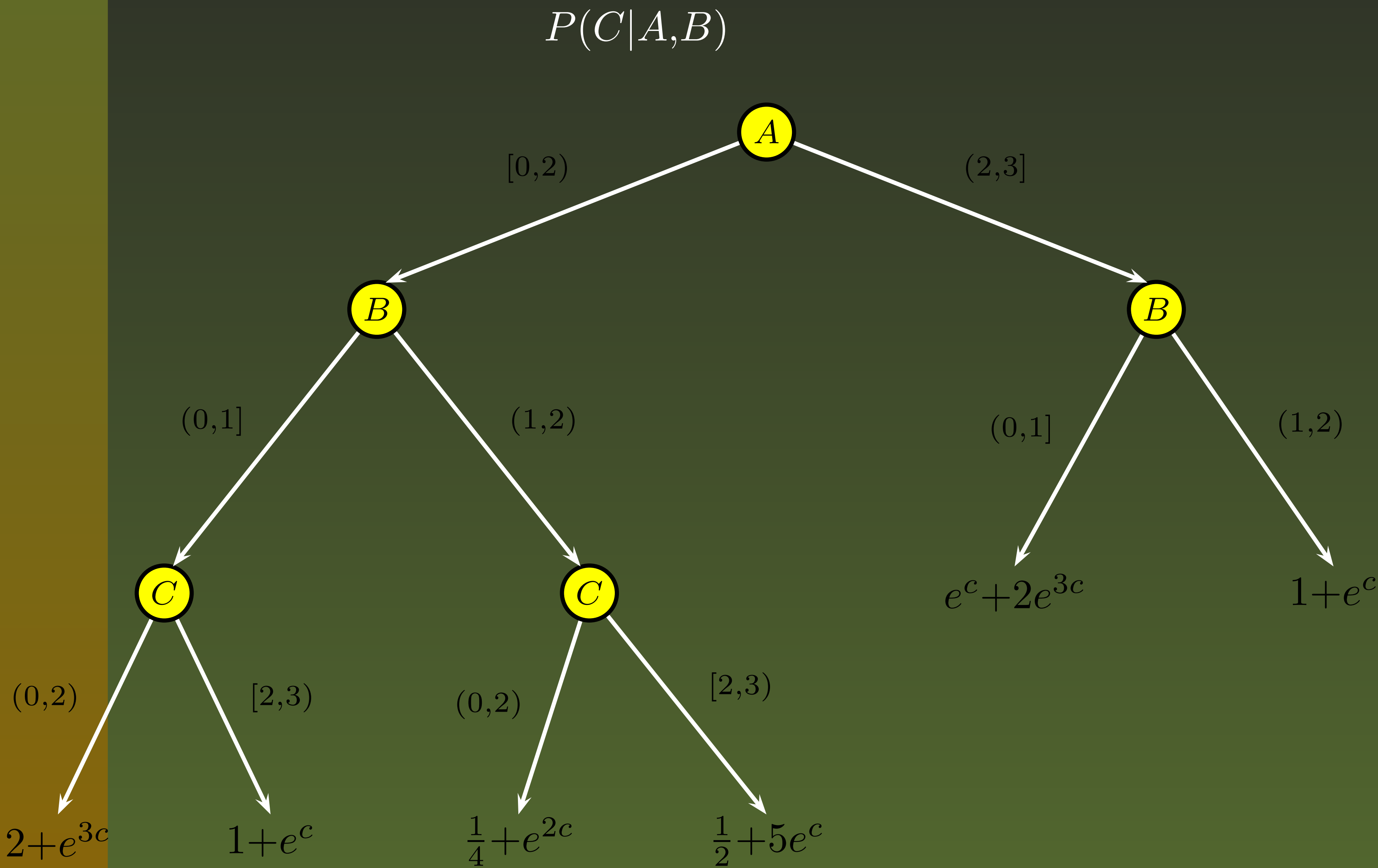
  II. $$\sum_{\mathbf{y}\in\Omega_{\mathbf{Y}}} \int_{\Omega_{\mathbf{Z}}} f(\mathbf{y},\mathbf{z})d\mathbf{z} = 1 \ .$$

- An MTE potential f defined over $\mathbf{X}_1 \cup \mathbf{X}_2$ is a conditional MTE density if:
  - For each $\mathbf{x}_2 \in \Omega_{\mathbf{X}_2}$, $f^{R(\mathbf{X}_2=\mathbf{x}_2)}$ is an MTE density for $\mathbf{X}_1$.

A data structure for MTE densities



Example of conditional MTE density



Constructing the tree

- Fixed number of splits.

Constructing the tree

- Fixed number of splits.
- Variables selected accorging to their splitting gain:

$$SG(C_i) = \sum_i e_i log(e_i)$$

Structural learning

- A mixed random vector $\mathbf{X} = \{X_1, \ldots, X_n\}$.

Structural learning

- A mixed random vector $\mathbf{X} = \{X_1, \ldots, X_n\}$.
- A sample of $\mathbf{X}$:

$$D = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\} \quad .$$

Structural learning

- A mixed random vector $\mathbf{X} = \{X_1, \ldots, X_n\}$.
- A sample of $\mathbf{X}$:

$$D = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\} \enspace .$$

- Goal: Find a Bayesian network G with variables $\mathbf{X}$, that agrees with the data D.

Issues in learning from data

- Determining a candidate structure G.

Issues in learning from data

- Determining a candidate structure G.
- Estimating the conditional distributions, $\hat{\theta}$, for G.

Issues in learning from data

- Determining a candidate structure G.
- Estimating the conditional distributions, $\hat{\theta}$, for G.
- Measuring the accuracy of $(G, \hat{\theta})$.

Searching for a structure

- Search in the space of possible networks.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.
- Movement operators:

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.
- Movement operators:
  - Insert an arc.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.
- Movement operators:
  - Insert an arc.
  - Delete an arc.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.
- Movement operators:
  - Insert an arc.
  - Delete an arc.
  - Invert an arc.

Searching for a structure

- Search in the space of possible networks.
- Hill climbing search.
- Starting point: network without arcs.
- Movement operators:
  - Insert an arc.
  - Delete an arc.
  - Invert an arc.
- After a movement, the affected conditional distributions are estimated.

Choosing the best move

Maximise a quality measure:

Choosing the best move

Maximise a quality measure:

$$Q(G|D, \hat{\theta})$$

Choosing the best move

Maximise a quality measure:

$$Q(G|D,\hat{\theta}) = \log L(D;G,\hat{\theta}) - \frac{1}{2}\mathrm{Dim}(G)\log m$$

Choosing the best move

Maximise a quality measure:

$$
\begin{aligned}
Q(G|D,\hat{\theta}) &= \log L(D;G,\hat{\theta}) - \frac{1}{2}\mathrm{Dim}(G)\log m \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} \log p_j(x_j^{(i)}|pa(x_j^{(i)})) - \frac{1}{2}\mathrm{Dim}(G)\log m
\end{aligned}
$$

Choosing the best move

Maximise a quality measure:

$$
\begin{aligned}
Q(G|D,\hat{\theta}) &= \log L(D;G,\hat{\theta}) - \frac{1}{2}\mathrm{Dim}(G)\log m \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\log p_j(x_j^{(i)}|pa(x_j^{(i)})) - \frac{1}{2}\mathrm{Dim}(G)\log m \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\log p_j(x_j^{(i)}|pa(x_j^{(i)})) - \frac{\log m}{2}\sum_{j=1}^{n}\prod_{X\in fa(X_j)}|X|
\end{aligned}
$$

Choosing the best move

$Q(G|D, \hat{\theta})$ is decomposable:

Choosing the best move

$Q(G|D,\hat{\theta})$ is decomposable:

$$Q(G|D,\hat{\theta})$$

Choosing the best move

$Q(G|D,\hat{\theta})$ is decomposable:

$$Q(G|D,\hat{\theta}) = \sum_{j=1}^{n} Q(X_j|D,\hat{\theta})$$

Choosing the best move

$Q(G|D, \hat{\theta})$ is decomposable:

$$Q(G|D, \hat{\theta}) = \sum_{j=1}^{n} Q(X_j|D, \hat{\theta})$$

$Q(X_j|D, \hat{\theta})$

Choosing the best move

$Q(G|D, \hat{\theta})$ is decomposable:

$$
\begin{aligned}
Q(G|D, \hat{\theta}) &= \sum_{j=1}^{n} Q(X_j|D, \hat{\theta}) \\
Q(X_j|D, \hat{\theta}) &= \sum_{i=1}^{m} \log p_j(x_j^{(i)}|pa(x_j^{(i)})) - \frac{\log m}{2} \prod_{X \in fa(X_j)} |X|
\end{aligned}
$$

Inference in MTE networks

- MTE potentials are closed for:

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.
  - Lazy propagation.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.
  - Lazy propagation.
  - Variable elimination.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.
  - Lazy propagation.
  - Variable elimination.
- Not valid methods (They require division):

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.
  - Lazy propagation.
  - Variable elimination.
- Not valid methods (They require division):
  - Hugin.

Inference in MTE networks

- MTE potentials are closed for:
  - Restriction.
  - Combination.
  - Marginalisation.
- Valid propagation algorithms:
  - Shenoy-Shafer.
  - Lazy propagation.
  - Variable elimination.
- Not valid methods (They require division):
  - Hugin.
  - Lauritzen-Spiegelhalter.

Penniless propagation

- High complexity: The size of the trees grow more quickly than in the discrete case.

Penniless propagation

- High complexity: The size of the trees grow more quickly than in the discrete case.
- Exact inference may become infeasible.

Penniless propagation

- High complexity: The size of the trees grow more quickly than in the discrete case.
- Exact inference may become infeasible.
- The trees are pruned in order to reduce their size.

Reducing the size of a tree

- Distance between two trees:

$$
\begin{aligned}
D(\mathcal{T},\mathcal{T}') &= E_{\phi_T}[(\phi_T - \phi_{T'})^2] \\
&= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}
\end{aligned}
$$

Reducing the size of a tree

- Distance between two trees:

$$
\begin{aligned}
D(\mathcal{T},\mathcal{T}') &= E_{\phi_T}[(\phi_T - \phi_{T'})^2] \\
&= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}
\end{aligned}
$$

- Removal of exponential terms.

Reducing the size of a tree

- Distance between two trees:

$$
\begin{aligned}
D(\mathcal{T},\mathcal{T}') &= E_{\phi_T}[(\phi_T - \phi_{T'})^2] \\
&= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}
\end{aligned}
$$

- Removal of exponential terms.
- Pruning leaves:

Reducing the size of a tree

- Distance between two trees:

$$D(\mathcal{T},\mathcal{T}') = E_{\phi_T}[(\phi_T - \phi_{T'})^2]$$
$$= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}$$

- Removal of exponential terms.
- Pruning leaves:
  - Joining children of a continuous variable.

Reducing the size of a tree

- Distance between two trees:

$$
\begin{aligned}
D(\mathcal{T},\mathcal{T}') &= E_{\phi_T}[(\phi_T - \phi_{T'})^2] \\
&= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}
\end{aligned}
$$

- Removal of exponential terms.
- Pruning leaves:
  - Joining children of a continuous variable.
  - Substituting a discrete variable by a number.

Reducing the size of a tree

- Distance between two trees:

$$
\begin{aligned}
D(\mathcal{T},\mathcal{T}') &= E_{\phi_T}[(\phi_T - \phi_{T'})^2] \\
&= \int_{\Omega_{\mathbf{z}}} \phi_T(\mathbf{z})(\phi_T(\mathbf{z}) - \phi_{T'}(\mathbf{z}))^2 d\mathbf{z}
\end{aligned}
$$

- Removal of exponential terms.
- Pruning leaves:
  - Joining children of a continuous variable.
  - Substituting a discrete variable by a number.
- Approximate whenever $D(\mathcal{T},\mathcal{T}') < \epsilon$.

Elimination of exponential terms

$$f(\mathbf{z}) = k + \sum_{i=1}^{n} a_i e^{b_i \mathbf{z}}$$

Elimination of exponential terms

$$f(\mathbf{z}) = k + \sum_{i=1}^{n} a_i e^{b_i \mathbf{z}} \rightarrow p_i = \int_{\Omega_{\mathbf{z}}} a_i e^{b_i \mathbf{z}} d\mathbf{z}$$

Elimination of exponential terms

$$f(\mathbf{z}) = k + \sum_{i=1}^{n} a_i e^{b_i \mathbf{z}} \rightarrow p_i = \int_{\Omega_{\mathbf{z}}} a_i e^{b_i \mathbf{z}} d\mathbf{z}$$

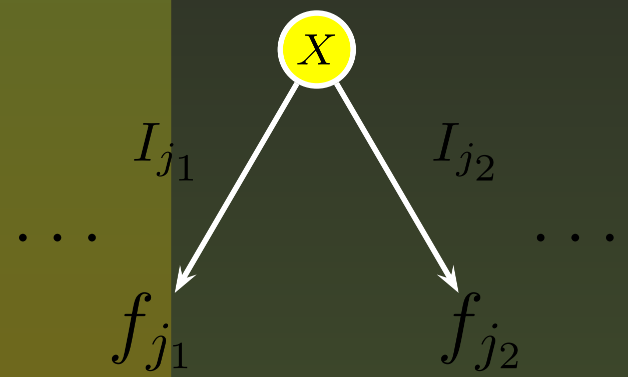
- Remove $a_i e^{b_i \mathbf{z}}$ if $|p_i| < \delta$

Elimination of exponential terms

$$f(\mathbf{z}) = k + \sum_{i=1}^{n} a_i e^{b_i \mathbf{z}} \rightarrow p_i = \int_{\Omega_{\mathbf{z}}} a_i e^{b_i \mathbf{z}} d\mathbf{z}$$

- Remove $a_i e^{b_i \mathbf{z}}$ if $|p_i| < \delta$
- Normalise f

Joining intervals



Joining intervals

Joining intervals

$$p_{j_1} = \int_{\Omega_{\mathbf{z}}} f_{j_1}(\mathbf{z}) d\mathbf{z} \quad and \quad p_{j_2} = \int_{\Omega_{\mathbf{z}}} f_{j_2}(\mathbf{z}) d\mathbf{z}$$

Joining intervals

$$p_{j_1} = \int_{\Omega_{\mathbf{z}}} f_{j_1}(\mathbf{z}) d\mathbf{z} \quad and \quad p_{j_2} = \int_{\Omega_{\mathbf{z}}} f_{j_2}(\mathbf{z}) d\mathbf{z}$$

$$f(\mathbf{z}) = \frac{p_{j_1} f_{j_1}(\mathbf{z}) + p_{j_2} f_{j_2}(\mathbf{z})}{p_{j_1} + p_{j_2}}$$

Joining intervals

$$p_{j_1} = \int_{\Omega_{\mathbf{z}}} f_{j_1}(\mathbf{z})d\mathbf{z} \quad and \quad p_{j_2} = \int_{\Omega_{\mathbf{z}}} f_{j_2}(\mathbf{z})d\mathbf{z}$$

$$f(\mathbf{z}) = \frac{p_{j_1} f_{j_1}(\mathbf{z}) + p_{j_2} f_{j_2}(\mathbf{z})}{p_{j_1} + p_{j_2}}$$

$$\int_{\Omega_{\mathbf{z}}} K f(\mathbf{z})d\mathbf{z} = p_{j_1} + p_{j_2}$$

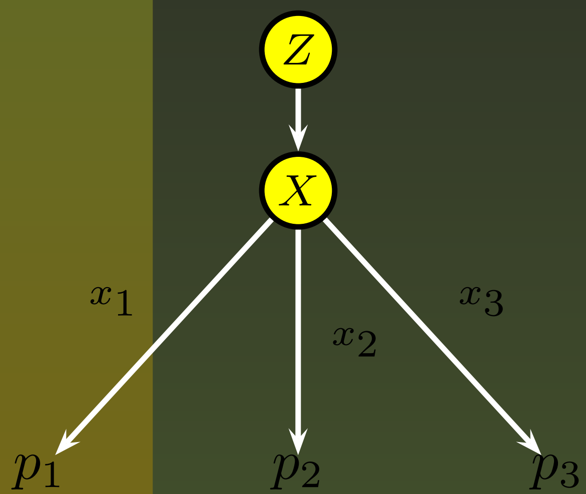
Joining intervals

$$p_{j_1} = \int_{\Omega_{\mathbf{z}}} f_{j_1}(\mathbf{z}) d\mathbf{z} \quad and \quad p_{j_2} = \int_{\Omega_{\mathbf{z}}} f_{j_2}(\mathbf{z}) d\mathbf{z}$$

$$f(\mathbf{z}) = \frac{p_{j_1} f_{j_1}(\mathbf{z}) + p_{j_2} f_{j_2}(\mathbf{z})}{p_{j_1} + p_{j_2}}$$

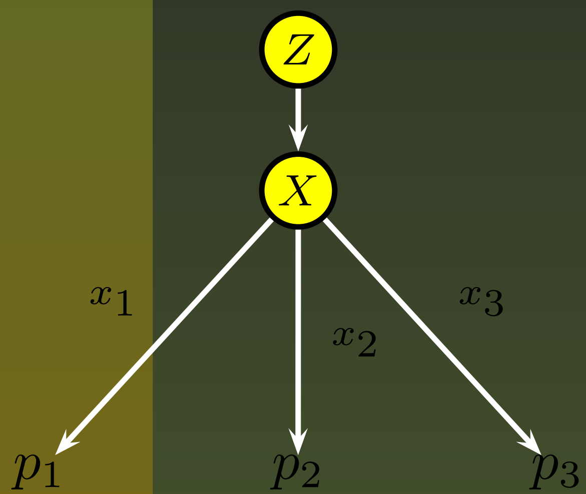
$$\int_{\Omega_{\mathbf{z}}} K f(\mathbf{z}) d\mathbf{z} = p_{j_1} + p_{j_2} \quad \Rightarrow \quad K = \frac{p_{j_1} + p_{j_2}}{\int_{\Omega_{\mathbf{z}}} f(\mathbf{z}) d\mathbf{z}}$$

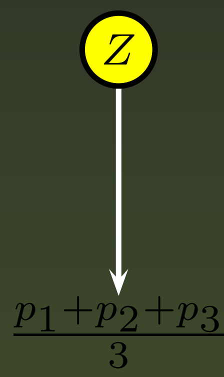
Pruning discrete variables



Pruning discrete variables

Pruning discrete variables





$$DK(p,q) = \sum_i p_i log(\frac{p_i}{q_i})$$

Pruning discrete variables

$$DK(p,q) = \sum_i p_i log(\frac{p_i}{q_i})$$

$$\xi = -((0.5-\epsilon)log(0.5-\epsilon) + (0.5+\epsilon)log(0.5+\epsilon))$$

Prune if $DK(\phi, \overline{p}) < \xi$

Propagation by simulation: MCMC

- Generate an initial configuration.

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:
  - Compute $P(X_i|\mathbf{W}_{X_i})$ restricted to the current configuration.

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:
  - Compute $P(X_i|\mathbf{W}_{X_i})$ restricted to the current configuration.
  - Simulate a value for X_i from it.

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:
  - Compute $P(X_i|\mathbf{W}_{X_i})$ restricted to the current configuration.
  - Simulate a value for X_i from it.
- Use the obtained sample to estimate the posterior distributions.

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:
  - Compute $P(X_i|\mathbf{W}_{X_i})$ restricted to the current configuration.
  - Simulate a value for X_i from it.
- Use the obtained sample to estimate the posterior distributions.
  - Estimate probabilities like $P(a < X < b)$.

Propagation by simulation: MCMC

- Generate an initial configuration.
- For each unobserved X_i:
  - Compute $P(X_i|\mathbf{W}_{X_i})$ restricted to the current configuration.
  - Simulate a value for X_i from it.
- Use the obtained sample to estimate the posterior distributions.
  - Estimate probabilities like $P(a < X < b)$.
  - Learn univariate densities from the sample.