Exact and Approximate Inference in Graphical Models
Variable Elimination and beyond

Nathalie Peyrard[a], Marie-José Cros[a], Simon de Givry[a], Alain Franc[b], Stéphane Robin[c], Régis Sabbadin[a], Thomas Schiex[a], Matthieu Vignes[a,d]

[a] INRA UR 875 MIAT, Toulouse, France
[b] INRA UMR 1202, BioGeCo, Bordeaux, France
[c] INRA-AgroParisTech, UMR 518 MIA, Paris, France
[d] IFS, Massey University, Palmerston North, New Zealand

JFRB'18, Toulouse, France
June, 1, 2018

Graphical models

- A graphical model over $\Lambda = \prod_{i=1}^{n} \Lambda_i$ is a decomposable function: $\psi : \prod_i \Lambda_i \to \mathbb{R}^+$ (the Λ_i are finite) which writes, $\forall x \in \Lambda$:

$$\psi(x) = \odot_{B \in \mathcal{B}} \psi_B(x_B),$$

where $\mathcal{B}$ is a set of subsets of $V = \{1, \ldots, n\}$
and $\odot \in \{\prod, \sum, \min, \max \ldots\}$ is a combination operator.

Graphical models

- A graphical model over $\Lambda = \prod_{i=1}^{n} \Lambda_i$ is a decomposable function: $\psi : \prod_i \Lambda_i \rightarrow \mathbb{R}^+$ (the Λ_i are finite) which writes, $\forall x \in \Lambda$:

$$\psi(x) = \odot_{B \in \mathcal{B}} \psi_B(x_B),$$

  where $\mathcal{B}$ is a set of subsets of $V = \{1, \ldots, n\}$
  and $\odot \in \{\prod, \sum, \min, \max \ldots\}$ is a combination operator.
- Used to model as well:
  - **Uncertainty**: (Hidden) Markov Chains, Bayesian networks, Markov Random Fields, Gaussian Graphical models, Possibilistic networks...
  - **Preferences**: (Weighted) Constraint Satisfaction Problems, Cost Function Networks...
  - **Both uncertainty and preferences**: Influence Diagrams...

Graphical models

- A graphical model over $\Lambda = \prod_{i=1}^{n} \Lambda_i$ is a decomposable function: $\psi : \prod_i \Lambda_i \rightarrow \mathbb{R}^+$ (the Λ_i are finite) which writes, $\forall x \in \Lambda$:

$$\psi(x) = \odot_{B \in \mathcal{B}} \psi_B(x_B),$$

  where $\mathcal{B}$ is a set of subsets of $V = \{1, \ldots, n\}$
  and $\odot \in \{\prod, \sum, \min, \max \ldots\}$ is a combination operator.
- Used to model as well:
  - **Uncertainty**: (Hidden) Markov Chains, Bayesian networks, Markov Random Fields, Gaussian Graphical models, Possibilistic networks...
  - **Preferences**: (Weighted) Constraint Satisfaction Problems, Cost Function Networks...
  - **Both uncertainty and preferences**: Influence Diagrams...
- In numerous application domains:
  - **Reasoning**: causal inference, information extraction...
  - **Computers**: Computer vision, speech recognition, LDPC codes...
  - **Bioinformatics**: Gene regulatory networks, protein structure...
  - **Environmental modelling**: Spatial and spatiotemporal processes...

Examples of probabilistic graphical models (1)

Definition (Markov chain)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.

$$P(x_1, \ldots, x_n) = \underbrace{P(x_1)}_{\psi_1(x_1)} \times \underbrace{P(x_2|x_1)}_{\psi_{12}(x_1,x_2)} \times \ldots \times \underbrace{P(x_n|x_{n-1})}_{\psi_{(n-1)n}(x_{n-1},x_n)}$$

Examples of probabilistic graphical models (1)

Definition (Markov chain)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.

$$P(x_1, \ldots, x_n) = \underbrace{P(x_1)}_{\psi_1(x_1)} \times \underbrace{P(x_2|x_1)}_{\psi_{12}(x_1,x_2)} \times \ldots \times \underbrace{P(x_n|x_{n-1})}_{\psi_{(n-1)n}(x_{n-1},x_n)}$$

Definition (Hidden Markov chain)

- $X = (X_1, \ldots, X_n)$ is an *unobserved* Markov chain.
- $Z = (Z_1, \ldots, Z_n)$ is a set of *observed* variables.
- $P(z|x) = \prod_{i=1}^{n} P(z_i|x_i)$ (independent observations).

$$P(x, z) = \underbrace{P(x_1)}_{\psi_1(x_1)} \times \left(\prod_{i=1}^{n-1} \underbrace{P(z_i|x_i)}_{\psi_i(x_i,z_i)} \times \underbrace{P(x_{i+1}|x_i)}_{\psi'_i(x_i,x_{i+1})} \right) \times \underbrace{P(z_n|x_n)}_{\psi_n(x_n,z_n)}$$

Examples of probabilistic graphical models (2)

Definition (Bayesian network)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.
- $Par(i) \subseteq \{1, \ldots, i-1\}, \forall i = 2, \ldots, n$.

$$P(x_1, \ldots, x_n) = \underbrace{P(x_1)}_{\psi_1(x_1)} \times \prod_{i=2}^{n} \underbrace{P(x_i | x_{Par(i)})}_{\psi_{Par(i)\cup\{i\}}(x_i, x_{Par(i)})}$$

Examples of probabilistic graphical models (2)

Definition (Bayesian network)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.
- $Par(i) \subseteq \{1, \ldots, i-1\}, \forall i = 2, \ldots, n$.

$$P(x_1, \ldots, x_n) = \underbrace{P(x_1)}_{\psi_1(x_1)} \times \prod_{i=2}^{n} \underbrace{P(x_i | x_{Par(i)})}_{\psi_{Par(i)\cup\{i\}}(x_i, x_{Par(i)})}$$

Definition (Markov Random Field)

- $G = (V, E)$ is an undirected graph with vertices $V = \{1, \ldots, n\}$, edges $E \in V \times V$ and $\mathcal{C}$ is the set of *cliques* of G.
- $\{\psi_C : X_C \to \mathbb{R}^{+*}\}_{C \in \mathcal{C}}$ are strictly positive functions.

$$P(x_1, \ldots, x_n) = \underbrace{\frac{1}{Z}}_{\psi_\emptyset, \text{ normalizing constant}} \times \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

Examples of non-probabilistic graphical models

Definition (Possibilistic networks)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.
- $Par(i) \subseteq \{1, \ldots, i-1\}, \forall i = 2, \ldots, n$.
- $\pi(X_1, \ldots, X_n)$ takes values in a finite totally ordered scale L.

$$\pi(x_1, \ldots, x_n) = \min \left\{ \underbrace{\pi(x_1)}_{\psi_1(x_1)}, \min_{i=2,\ldots,n} \underbrace{\pi(x_i | x_{Par(i)})}_{\psi_{Par(i)\cup\{i\}}(x_i, x_{Par(i)})} \right\}$$

Examples of non-probabilistic graphical models

Definition (Possibilistic networks)

- $X = (X_1, \ldots, X_n)$ is a set of variables, with finite domains $\{\Lambda_i\}_{i=1,\ldots,n}$.
- $Par(i) \subseteq \{1, \ldots, i-1\}, \forall i = 2, \ldots, n$.
- $\pi(X_1, \ldots, X_n)$ takes values in a finite totally ordered scale L.

$$\pi(x_1, \ldots, x_n) = \min \left\{ \underbrace{\pi(x_1)}_{\psi_1(x_1)}, \min_{i=2,\ldots,n} \underbrace{\pi(x_i | x_{Par(i)})}_{\psi_{Par(i)\cup\{i\}}(x_i, x_{Par(i)})} \right\}$$

Definition (Cost Functions networks)

- $\{w_C : X_C \to \mathbb{R}^+\}_{C \in \mathcal{C}}$ are positive functions.

$$w(x_1, \ldots, x_n) = \sum_{c \in \mathcal{C}} w_C(x_C)$$

More complex graphical models

Some graphical models can have more than one combination operator:

Markov decision process / Influence Diagram

- Markov chain/Bayesian network *plus* decision variables *plus* cost functions network
- Expected utility function

$\rightarrow$ We will limit ourselves to single operator graphical models!

Graphical model definition

Definition (Graphical model)

- Let $X = (X_1, \ldots, X_n)$ be a set of variables.
- X_i takes values in $\Lambda_i \subseteq \mathbb{R}$.
- A realization of X is denoted $x = (x_1, \ldots, x_n)$, with $x_i \in \Lambda_i$.
- A graphical model over X is a function $\psi : \prod_i \Lambda_i \to \mathbb{R}$, which writes, $\forall x \in X$:

$$\psi(x) = \odot_{B \in \mathcal{B}} \psi_B(x_B),$$

where $\mathcal{B}$ is a set of subsets of $V = \{1, \ldots, n\}$, $\psi_B : \prod_{i \in B} \Lambda_i \to \mathbb{R}$ and $\odot \in \{\prod, \sum, \min, \max ...\}$ is a combination operator.

Why are these models called "graphical"?

They admit "graphical" representations:

- Useful for variables interactions vizualization
- Allow to directly extract computational features:
  Treewidth, perfect elimination ordering...
- Graphical properties sometimes exploited directly in algorithms

Why are these models called "graphical"?

Graphical representations of a graphical model:

- Directed graph representation
  $\Rightarrow$ Represent conditional dependence between variables...
- Undirected graph representation
  $\Rightarrow$ Represent conditional indepence between variables...
  And useful to compute variable elimination orderings
- Hypergraph representation
  $\Rightarrow$ Directly represents the graphical model functions scopes
- Factor graph representation
  $\Rightarrow$ Equivalent to hypergraph representation.
  Useful in message passing algorithms

Illustration on a bayesian network

$$P(X_1, \ldots, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1, X_3)P(X_5|X_2, X_4)$$

Illustration on a bayesian network

$$P(X_1, \ldots, X_5) \quad = \quad P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1, X_3)P(X_5|X_2, X_4)$$

Directed acyclic graph

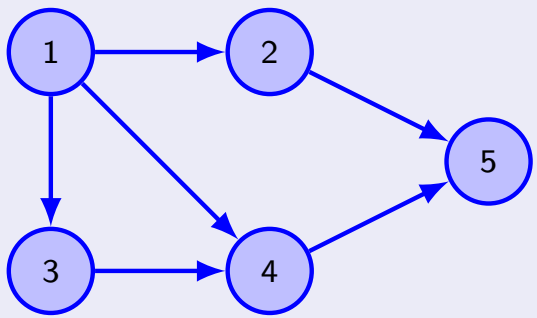
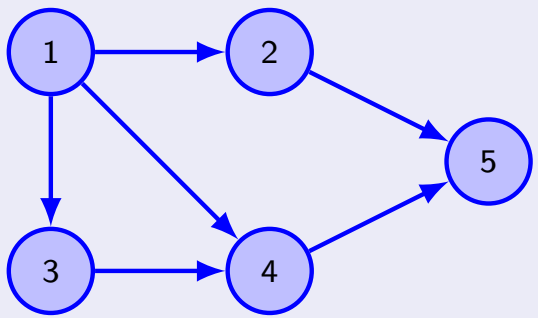


Illustration on a bayesian network

$$P(X_1, \ldots, X_5) \quad = \quad P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1, X_3)P(X_5|X_2, X_4)$$

Directed acyclic graph



Undirected graph

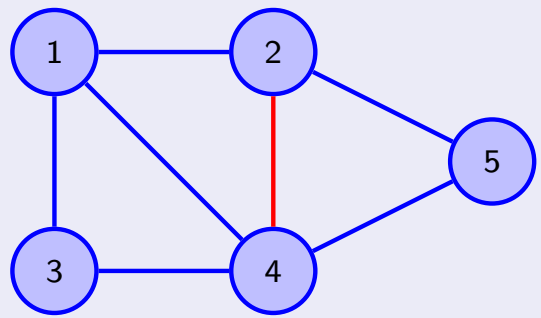
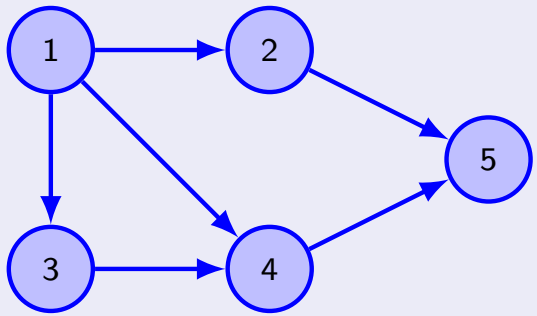


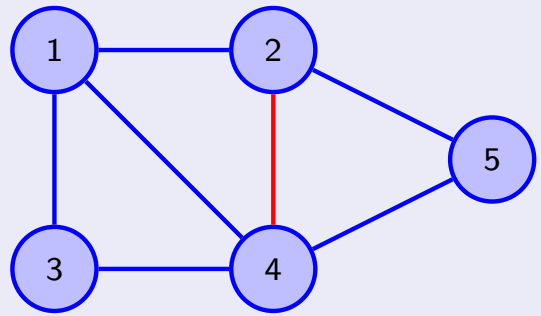
Illustration on a bayesian network

$$
\begin{aligned}
P(X_1,\dots,X_5) &= P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1,X_3)P(X_5|X_2,X_4)\\
\psi_{12}(X_1,X_2) &= P(X_1)P(X_2|X_1)\\
\psi_{134}(X_1,X_3,X_4) &= P(X_3|X_1)P(X_4|X_1,X_3)\ ;\ \psi_{245}(X_2,X_4,X_5) = P(X_5|X_2,X_4)
\end{aligned}
$$

Directed acyclic graph



Undirected graph



Hypergraph

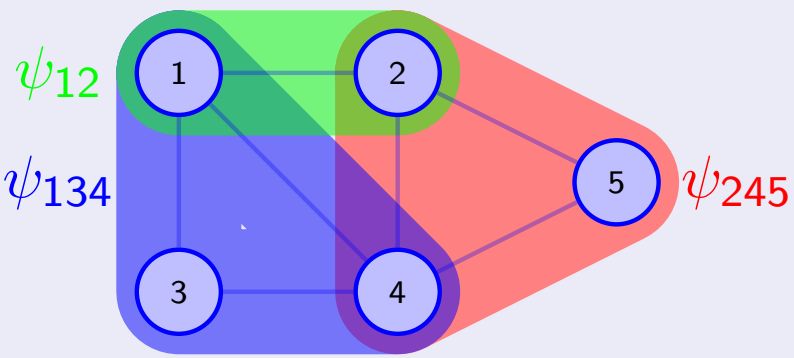
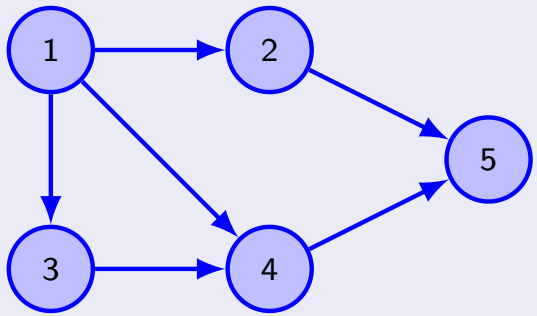


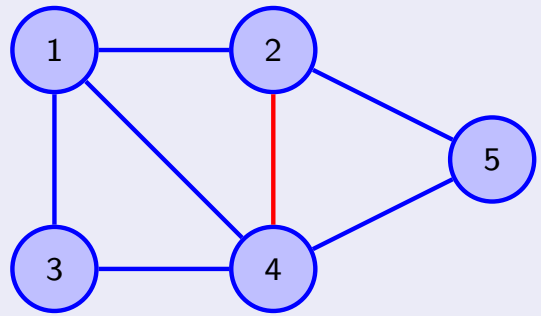
Illustration on a bayesian network

$$
\begin{aligned}
P(X_1,\dots,X_5) &= P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_1,X_3)P(X_5|X_2,X_4)\\
\psi_{12}(X_1,X_2) &= P(X_1)P(X_2|X_1)\\
\psi_{134}(X_1,X_3,X_4) &= P(X_3|X_1)P(X_4|X_1,X_3)\ ;\ \psi_{245}(X_2,X_4,X_5) = P(X_5|X_2,X_4)
\end{aligned}
$$

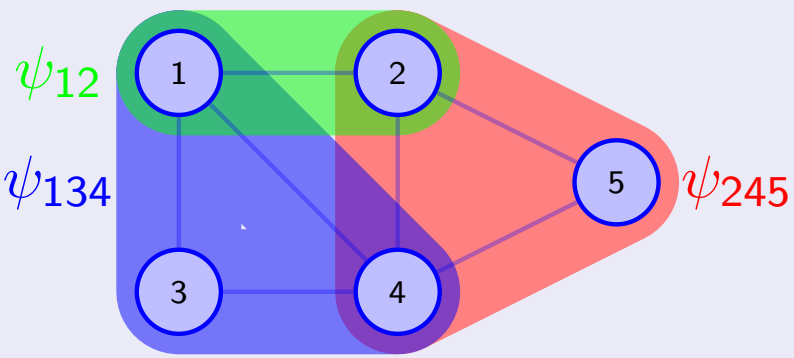
Directed acyclic graph



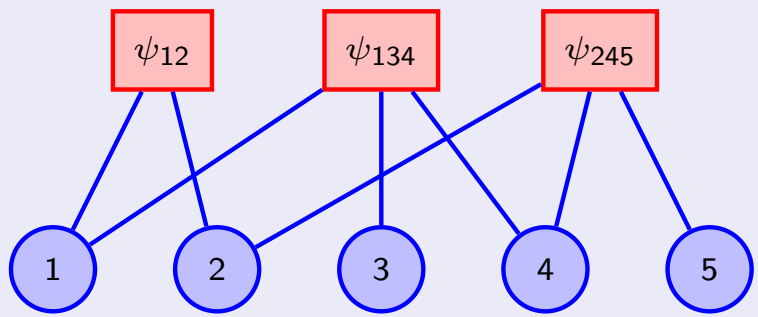
Undirected graph



Hypergraph



Factor graph



Inference tasks in graphical models

There are basically two usual inference tasks that can be solved using graphical models:

- Optimizing:

Most likely solutions in stochastic graphical models

$$x^* = \arg\max_x P(x_1, \ldots, x_n)$$

Most likely solutions in possibilistic networks

$$x^* = \arg\max_x \pi(x_1, \ldots, x_n)$$

min-cost solutions in cost function networks

$$x^* = \arg\min_x w(x_1, \ldots, x_n)$$

Inference tasks in graphical models

There are basically two usual inference tasks that can be solved using graphical models:

- Optimizing
- Counting:

Marginal probabilities computation in stochastic graphical models

$$P(x_B) = \sum_{x_{\bar{B}}} P(x_B, x_{\bar{B}})$$

Normalizing constant computation in Markov Random Fields

$$Z = \sum_{x} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

Solution counting in constraint satisfaction problems

$$\#SAT = |\{x, w(x) = 0\}|$$

Inference tasks in graphical models

There are basically two usual inference tasks that can be solved using graphical models:

- Optimizing
- Counting

These are sometimes interleaved, in more complex problems:

Maximum Expected Utility computation in influence diagrams

$$MEU = \max_{d_{D_1}} \sum_{x_{S_1}} \dots \max_{d_{D_k}} \sum_{x_{S_k}} P(\underbrace{x_1, \dots, x_n}_{x} \mid \underbrace{d_1, \dots, d_m}_{d}) U(x, d)$$

Generalized Quantified Boolean Formula satisfiability

$$UnSAT = \min_{x_{A_1}} \max_{y_{B_1}} \dots \min_{x_{A_k}} \max_{y_{B_k}} w(\underbrace{x_1, \dots, x_n}_{x}, \underbrace{y_1, \dots, y_m}_{y})$$

$\rightarrow$ Some of the following inference approaches may be used as well for these more complex tasks!

Inference approaches

- Exact inference
  - Exhaustive exploration (optimizing, solution finding, counting)
  - Heuristic search (optimizing, solution finding)
  - Variable elimination (counting, optimizing, solution finding)
- Approximate inference
  - Sampling based approaches:
    counting, optimizing, both (e.g. reinforcement learning)...
  - Heuristic-based approaches
  - Loopy belief-propagation (approximate counting or optimizing)
  - Variational approximation (stochastic models)

Variable elimination, case of Hidden Markov Chains

Problem

Computing the most likely values of variables H given a realization o of the variables O. The problem is to compute $\arg\max_h p(H = h|O = o)$, or equivalently the argument of:

$$\max_{h_1,\ldots,h_T} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-1} \psi_{H_i,H_{i+1}}(h_i, h_{i+1}) \psi_{O_i,H_i}(o_i, h_i) \right) \psi_{O_T,H_T}(o_T, h_T)$$

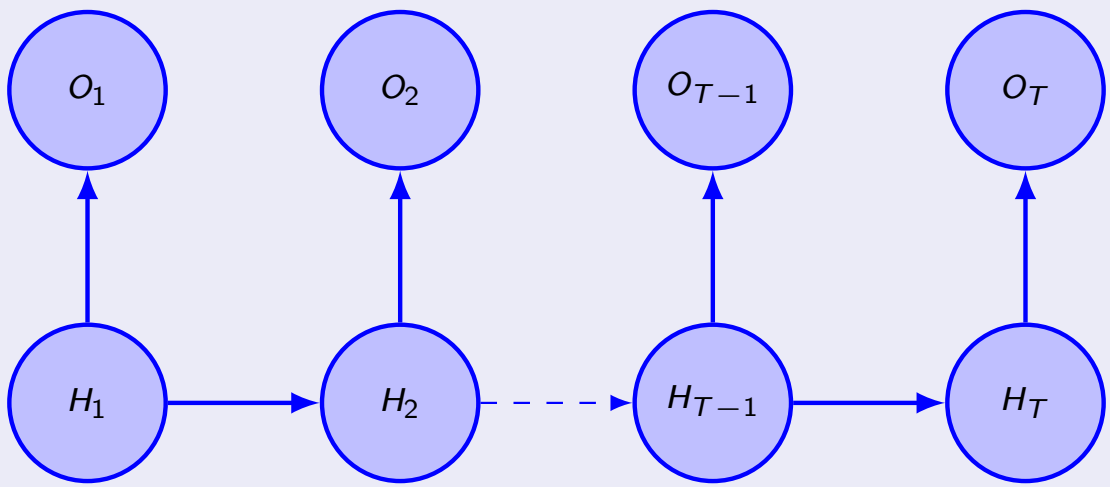
Remarks:

- The number of possible realizations of H is exponential in T
- Nevertheless, this optimization problem can be solved in a number of operations linear in T using the well-known Viterbi algorithm

Variable elimination in HMC: Viterbi algorithm

$$\max_{h_1,\dots,h_T} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-1} \psi_{H_i,H_{i+1}}(h_i, h_{i+1}) \psi_{O_i,H_i}(o_i, h_i) \right) \psi_{O_T,H_T}(o_T, h_T)$$

Viterbi algorithm

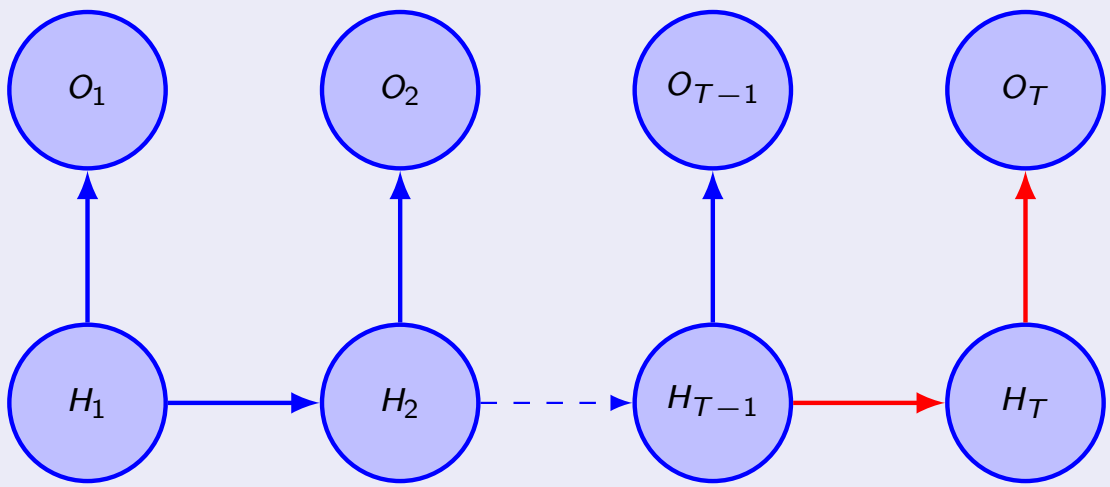


Variable elimination in HMC: Viterbi algorithm

$$= \max_{h_1,\dots,h_{T-1}} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-2} \psi_{H_i,H_{i+1}}(h_i, h_{i+1}) \psi_{O_i,H_i}(o_i, h_i) \right)$$

$$\times \psi_{O_{T-1},H_{T-1}}(o_{T-1}, h_{T-1}) \times \underbrace{\max_{h_T} \psi_{H_{T-1},H_T}(h_{T-1}, h_T) \psi_{O_T,H_T}(o_T, h_T)}_{\text{New potential function } \psi'_{O_T,H_{T-1}}(o_T, h_{T-1})}$$

Viterbi algorithm

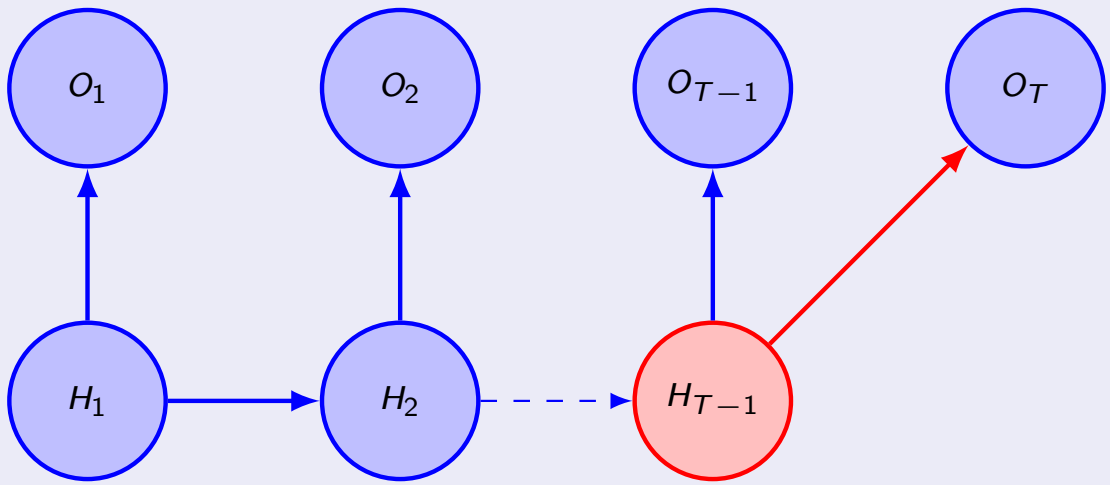


Variable elimination in HMC: Viterbi algorithm

$$= \max_{h_1,\dots,h_{T-1}} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-2} \psi_{H_i,H_{i+1}}(h_i, h_{i+1}) \psi_{O_i,H_i}(o_i, h_i) \right)$$

$$\times \psi_{O_{T-1},H_{T-1}}(o_{T-1}, h_{T-1}) \times \underbrace{\max_{h_T} \psi_{H_{T-1},H_T}(h_{T-1}, h_T) \psi_{O_T,H_T}(o_T, h_T)}_{\text{New potential function } \psi'_{O_T,H_{T-1}}(o_T, h_{T-1})}$$

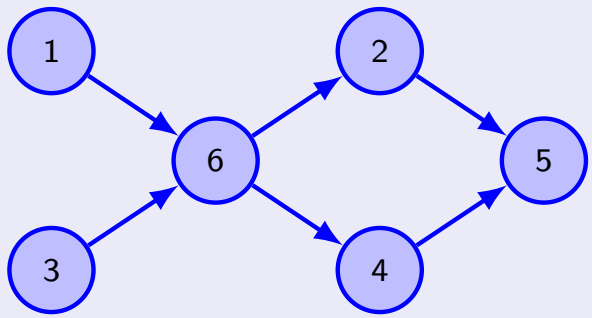
Viterbi algorithm



Variable elimination, bayesian network

$$P(X_1, \ldots, X_6) = P(X_1)P(X_3)P(X_6|X_1, X_3)P(X_2|X_6)P(X_4|X_6)P(X_5|X_2, X_4)$$

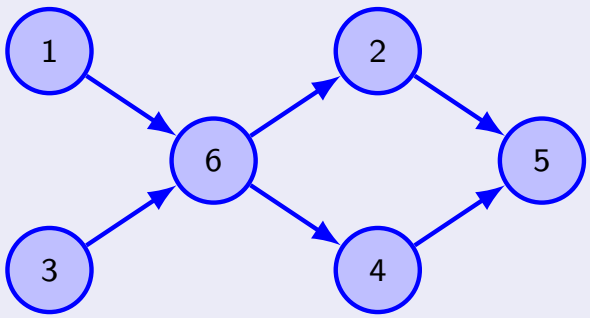
Directed graph



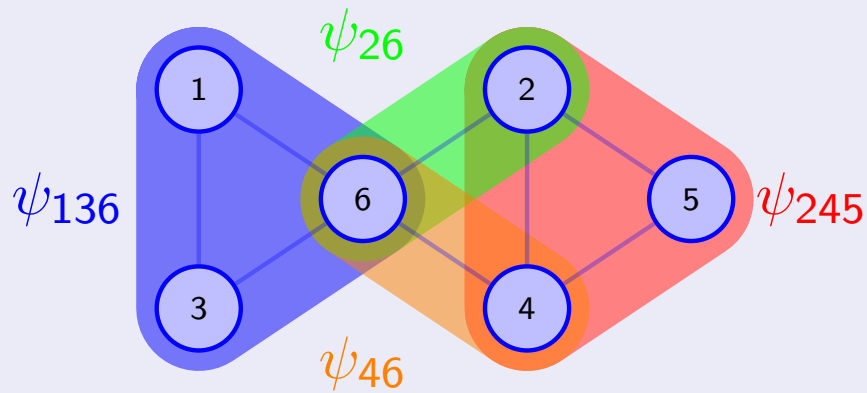
Variable elimination, bayesian network

$$
\begin{aligned}
P(X_1,\dots,X_6) &= P(X_1)P(X_3)P(X_6|X_1,X_3)P(X_2|X_6)P(X_4|X_6)P(X_5|X_2,X_4)\\
P(X_1,\dots,X_6) &= \psi_{136}(X_1,X_3,X_6)\psi_{26}(X_2,X_6)\psi_{46}(X_4,X_6)\psi_{245}(X_2,X_4,X_5)\\
P(X_1,\dots,X_6) &= \psi_{136}\psi_{26}\psi_{46}\psi_{245}
\end{aligned}
$$

Directed graph



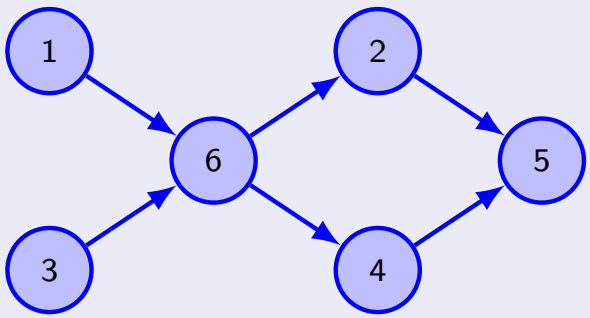
Undirected graph / Hypergraph



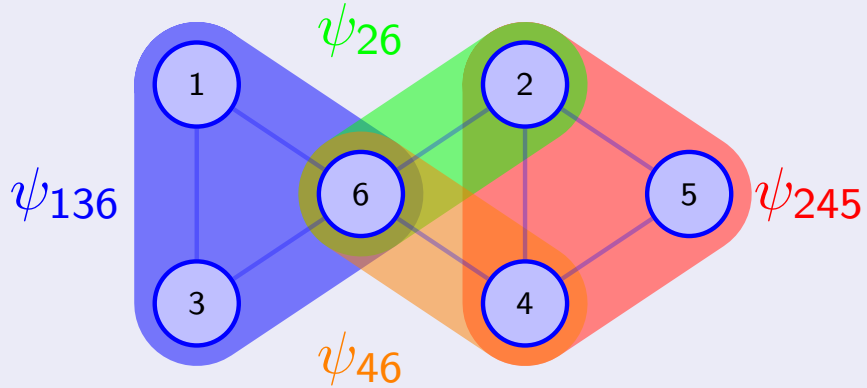
Variable elimination, bayesian network

$$
\begin{aligned}
P(X_1,\dots,X_6) &= P(X_1)P(X_3)P(X_6|X_1,X_3)P(X_2|X_6)P(X_4|X_6)P(X_5|X_2,X_4)\\
P(X_1,\dots,X_6) &= \psi_{136}(X_1,X_3,X_6)\psi_{26}(X_2,X_6)\psi_{46}(X_4,X_6)\psi_{245}(X_2,X_4,X_5)\\
P(X_1,\dots,X_6) &= \psi_{136}\psi_{26}\psi_{46}\psi_{245}
\end{aligned}
$$

Directed graph



Undirected graph / Hypergraph



Compute $\alpha = \max_{X_1,\dots,X_6} P(X_1,\dots,X_6) = \max_{X_1,\dots,X_6} \psi_{136}\psi_{26}\psi_{46}\psi_{245}$

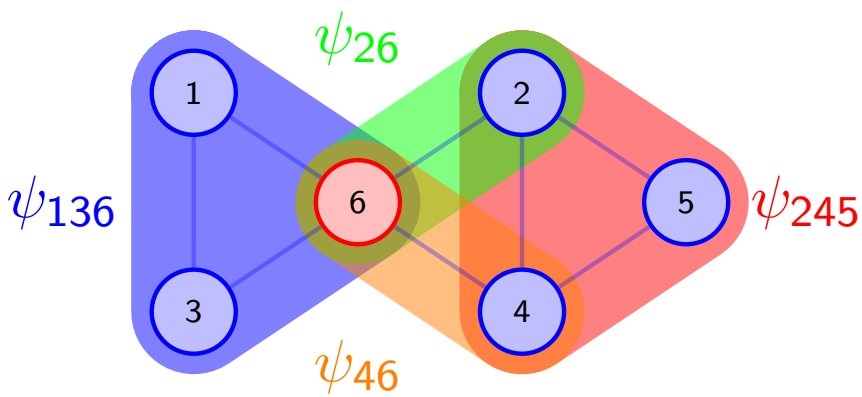
Elimination ordering

- Variable elimination, for elimination order $\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}$:

$$\alpha = \max_{X_1,\dots,X_6} \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5)$$

$$\alpha = \max_{X_1,\dots,X_5} \psi_{245} \max_{X_6} \psi_{136}\psi_{26}\psi_{46} = \max_{X_1,\dots,X_5} \psi_{245}\psi'_{1234}$$

$$\alpha = \max_{X_1,\dots,X_4} \psi'_{1234} \max_{X_5} \psi_{245} = \max_{X_1,\dots,X_4} \psi'_{1234}\psi'_{24} = \max_{X_1,\dots,X_4} \psi''_{1234}$$



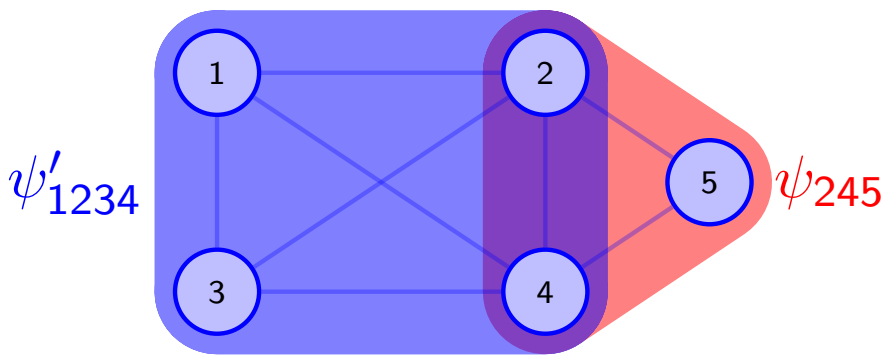
Elimination ordering

- Variable elimination, for elimination order $\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}$:

$$\alpha = \max_{X_1,\dots,X_6} \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5)$$

$$\alpha = \max_{X_1,\dots,X_5} \psi_{245} \max_{X_6} \psi_{136}\psi_{26}\psi_{46} = \max_{X_1,\dots,X_5} \psi_{245}\psi'_{1234}$$

$$\alpha = \max_{X_1,\dots,X_4} \psi'_{1234} \max_{X_5} \psi_{245} = \max_{X_1,\dots,X_4} \psi'_{1234}\psi'_{24} = \max_{X_1,\dots,X_4} \psi''_{1234}$$



Elimination ordering

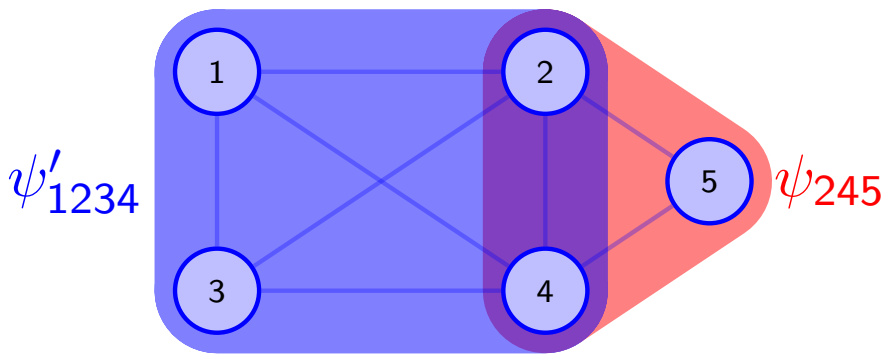
- Variable elimination, for elimination order $\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}$:

$$\alpha = \max_{X_1,\dots,X_6} \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5)$$

$$\alpha = \max_{X_1,\dots,X_5} \psi_{245} \max_{X_6} \psi_{136}\psi_{26}\psi_{46} = \max_{X_1,\dots,X_5} \psi_{245}\psi'_{1234}$$

$$\alpha = \max_{X_1,\dots,X_4} \psi'_{1234} \max_{X_5} \psi_{245} = \max_{X_1,\dots,X_4} \psi'_{1234}\psi'_{24} = \max_{X_1,\dots,X_4} \psi''_{1234}$$

  - Is there an elimination order generating "smaller scopes" functions?



Elimination ordering

- Variable elimination, for elimination order $\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}$:

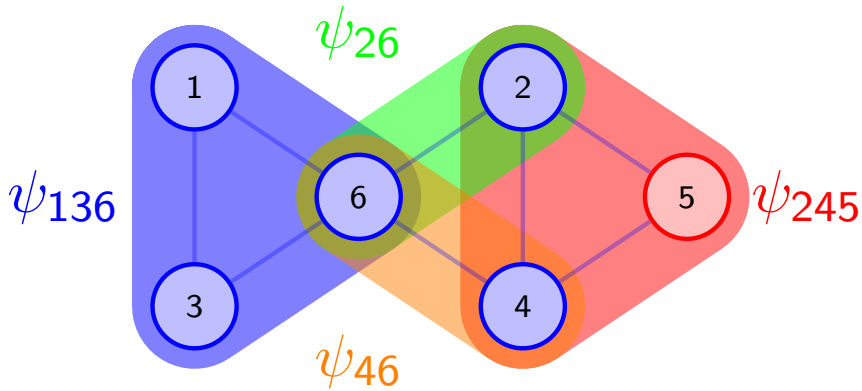
$$\begin{aligned}
\alpha &= \max_{X_1,\dots,X_6} \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5) \\
\alpha &= \max_{X_1,\dots,X_5} \psi_{245} \max_{X_6} \psi_{136}\psi_{26}\psi_{46} = \max_{X_1,\dots,X_5} \psi_{245}\psi'_{1234} \\
\alpha &= \max_{X_1,\dots,X_4} \psi'_{1234} \max_{X_5} \psi_{245} = \max_{X_1,\dots,X_4} \psi'_{1234}\psi'_{24} = \max_{X_1,\dots,X_4} \psi''_{1234}
\end{aligned}$$

  - Is there an elimination order generating "smaller scopes" functions?

- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

$$\begin{aligned}
\alpha &= \max_{X_1,X_3,X_6,X_2,X_4,X_5} \psi_{136}\psi_{26}\psi_{46}\psi_{245} \\
\alpha &= \max_{X_1,\dots,X_4} \psi_{136}\psi_{26}\psi_{46} \max_{X_5} \psi_{245} = \max_{X_1,\dots,X_4} \psi_{136}\psi_{26}\psi_{46}\psi'_{24} \\
\alpha &= \max_{X_1,\dots,X_2} \psi_{136} \max_{X_4} \psi'_{246} = \max_{X_1,X_3,X_6} \psi_{136} \max_{X_2} \psi'_{26} = \max_{X_1,X_3,X_6} \psi'_{136}
\end{aligned}$$

Elimination ordering



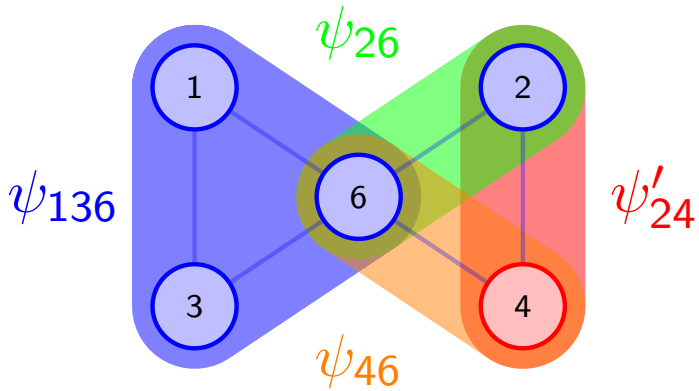
- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

$$\alpha = \max_{X_1,X_3,X_6,X_2,X_4,X_5} \psi_{136}\psi_{26}\psi_{46}\psi_{245}$$

$$\alpha = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46} \max_{X_5} \psi_{245} = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46}\psi'_{24}$$

$$\alpha = \max_{X_1,\ldots,X_2} \psi_{136} \max_{X_4} \psi'_{246} = \max_{X_1,X_3,X_6} \psi_{136} \max_{X_2} \psi'_{26} = \max_{X_1,X_3,X_6} \psi'_{136}$$

Elimination ordering



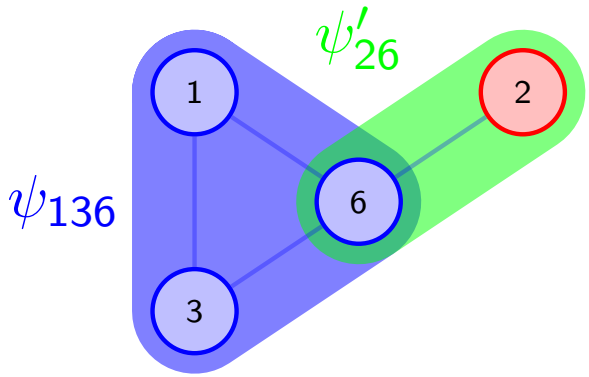
- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

$$\alpha = \max_{X_1,X_3,X_6,X_2,X_4,X_5} \psi_{136}\psi_{26}\psi_{46}\psi_{245}$$

$$\alpha = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46} \max_{X_5} \psi_{245} = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46}\psi'_{24}$$

$$\alpha = \max_{X_1,\ldots,X_2} \psi_{136} \max_{X_4} \psi'_{246} = \max_{X_1,X_3,X_6} \psi_{136} \max_{X_2} \psi'_{26} = \max_{X_1,X_3,X_6} \psi'_{136}$$

Elimination ordering



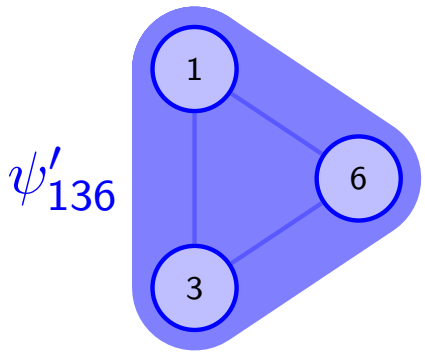
- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

$$\alpha = \max_{X_1,X_3,X_6,X_2,X_4,X_5} \psi_{136}\psi_{26}\psi_{46}\psi_{245}$$

$$\alpha = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46} \max_{X_5} \psi_{245} = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46}\psi'_{24}$$

$$\alpha = \max_{X_1,\ldots,X_2} \psi_{136} \max_{X_4} \psi'_{246} = \max_{X_1,X_3,X_6} \psi_{136} \max_{X_2} \psi'_{26} = \max_{X_1,X_3,X_6} \psi'_{136}$$

Elimination ordering



- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

$$\begin{aligned}
\alpha &= \max_{X_1,X_3,X_6,X_2,X_4,X_5} \psi_{136}\psi_{26}\psi_{46}\psi_{245} \\
\alpha &= \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46} \max_{X_5} \psi_{245} = \max_{X_1,\ldots,X_4} \psi_{136}\psi_{26}\psi_{46}\psi'_{24} \\
\alpha &= \max_{X_1,\ldots,X_2} \psi_{136} \max_{X_4} \psi'_{246} = \max_{X_1,X_3,X_6} \psi_{136} \max_{X_2} \psi'_{26} = \max_{X_1,X_3,X_6} \psi'_{136}
\end{aligned}$$

$\Rightarrow$ π generates functions with 4 variables, instead of 3 at most for π^*!

Treewidth

Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Variable/vertex elimination)

Let $i \in \{1, \ldots, n\}$ be a vertex/variable number. We define:

$$\mathcal{H}^{\downarrow i} = \{H_j \in \mathcal{H}, i \notin H_j\} \cup \Big\{ \bigcup_{i \in H_j \in \mathcal{H}} H_j \setminus \{i\} \Big\}.$$

Treewidth

Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Variable/vertex elimination)

Let $i \in \{1, \ldots, n\}$ be a vertex/variable number. We define:

$$\mathcal{H}^{\downarrow i} = \{H_j \in \mathcal{H}, i \notin H_j\} \cup \Big\{ \bigcup_{i \in H_j \in \mathcal{H}} H_j \setminus \{i\} \Big\}.$$

Definition (Induced width)

- Let π be an elimination order ($\pi(k)$ is the k^{th} eliminated vertex).
- Let $\mathcal{H}^k$ be the hypergraph generated after k vertices eliminations ($\mathcal{H}^0 = \mathcal{H}$ and $\mathcal{H}^n = \emptyset$).

$\Rightarrow$ The induced width is the size of the largest generated hyperedge (minus 1), during the process :

$$\mathcal{I}W_\pi(\mathcal{H}) = \max_{k=0,..,n-1} \max_{H \in \mathcal{H}^k} |H| - 1.$$

Treewidth

Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

$\mathcal{TW}(\mathcal{H})$, the treewidth of $\mathcal{H}$ is the minimum induced width over all elimination orders:

$$\mathcal{TW}(\mathcal{H}) = \min_{\pi} \mathcal{IW}_{\pi}(\mathcal{H}).$$

Treewidth

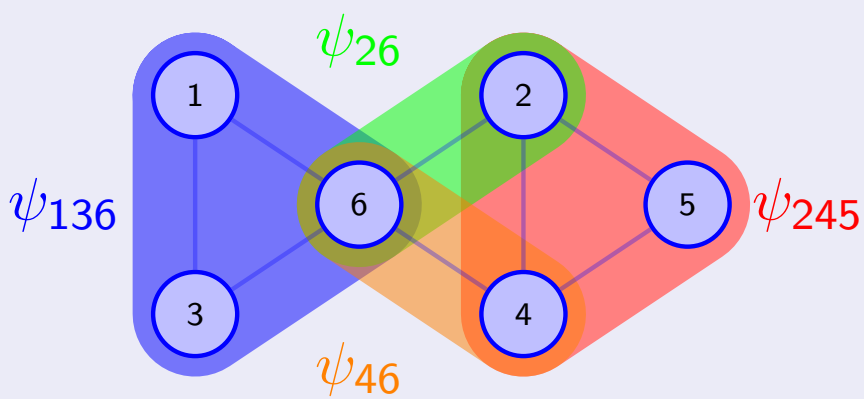
Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

$\mathcal{TW}(\mathcal{H})$, the treewidth of $\mathcal{H}$ is the minimum induced width over all elimination orders:

$$\mathcal{TW}(\mathcal{H}) = \min_{\pi} \mathcal{IW}_{\pi}(\mathcal{H}).$$

Undirected graph / Hypergraph



Treewidth

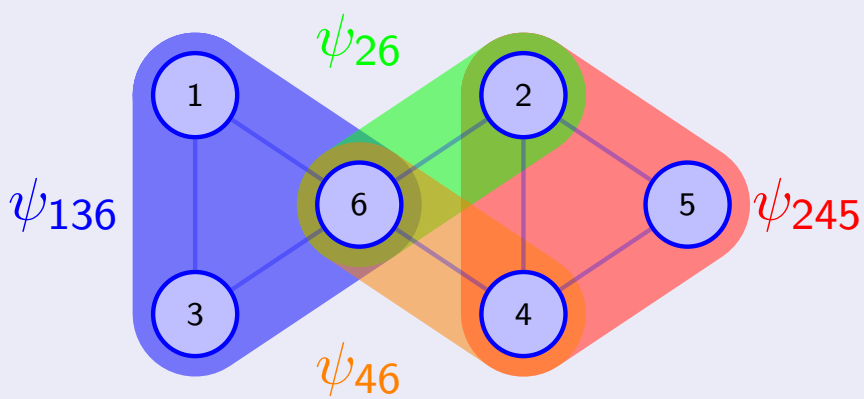
Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

$\mathcal{TW}(\mathcal{H})$, the treewidth of $\mathcal{H}$ is the minimum induced width over all elimination orders:

$$\mathcal{TW}(\mathcal{H}) = \min_{\pi} \mathcal{IW}_{\pi}(\mathcal{H}).$$

Undirected graph / Hypergraph



- $\pi = \{6, 5, 4, 3, 2, 1\}$
- $\mathcal{IW}_{\pi}(\mathcal{H}) = 3$

Treewidth

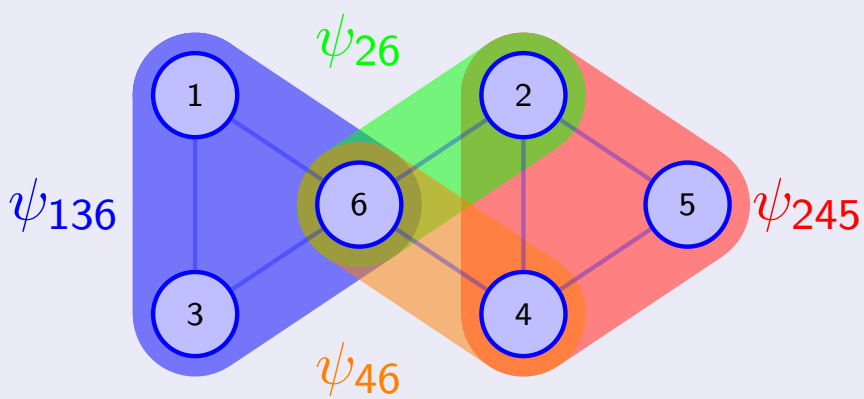
Let $\mathcal{H} = \{H_1, \ldots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

$\mathcal{TW}(\mathcal{H})$, the treewidth of $\mathcal{H}$ is the minimum induced width over all elimination orders:

$$\mathcal{TW}(\mathcal{H}) = \min_{\pi} \mathcal{IW}_{\pi}(\mathcal{H}).$$

Undirected graph / Hypergraph



- $\pi = \{6, 5, 4, 3, 2, 1\}$
- $\mathcal{IW}_{\pi}(\mathcal{H}) = 3$
- $\pi^* = \{5, 2, 4, 6, 3, 1\}$
- $\mathcal{TW}(\mathcal{H}) = \mathcal{IW}_{\pi^*}(\mathcal{H}) = 2$

Treewidth computation

Variable elimination takes exponential space/time in induced width

$\Rightarrow$ Important to find a "good" elimination order (small induced width)

Treewidth computation

Variable elimination takes exponential space/time in induced width

$\Rightarrow$ Important to find a "good" elimination order (small induced width)

Determining the treewidth is a hard problem

- Computing the treewidth is NP-hard
- Approximating the treewidth with constant factor?

$\Rightarrow$ Not known whether in P or NP-hard

Treewidth computation

Variable elimination takes exponential space/time in induced width

$\Rightarrow$ Important to find a "good" elimination order (small induced width)

Determining the treewidth is a hard problem

- Computing the treewidth is NP-hard
- Approximating the treewidth with constant factor?

$\Rightarrow$ Not known whether in P or NP-hard

Treewidth computation is a **Fixed Parameter Tractable** problem

- An $O\left(\mathcal{TW}(\mathcal{H})^{\mathcal{TW}(\mathcal{H})^3} n\right)$ solution algorithm (impractical)

Treewidth computation

Variable elimination takes exponential space/time in induced width

$\Rightarrow$ Important to find a "good" elimination order (small induced width)

Determining the treewidth is a hard problem

- Computing the treewidth is NP-hard
- Approximating the treewidth with constant factor?

$\Rightarrow$ Not known whether in P or NP-hard

Treewidth computation is a **Fixed Parameter Tractable** problem

- An $O\left(\mathcal{TW}(\mathcal{H})^{\mathcal{TW}(\mathcal{H})^3} n\right)$ solution algorithm (impractical)

$\Rightarrow$ In practice, elimination order heuristics (e.g. min-fill).

From variable elimination to message passing

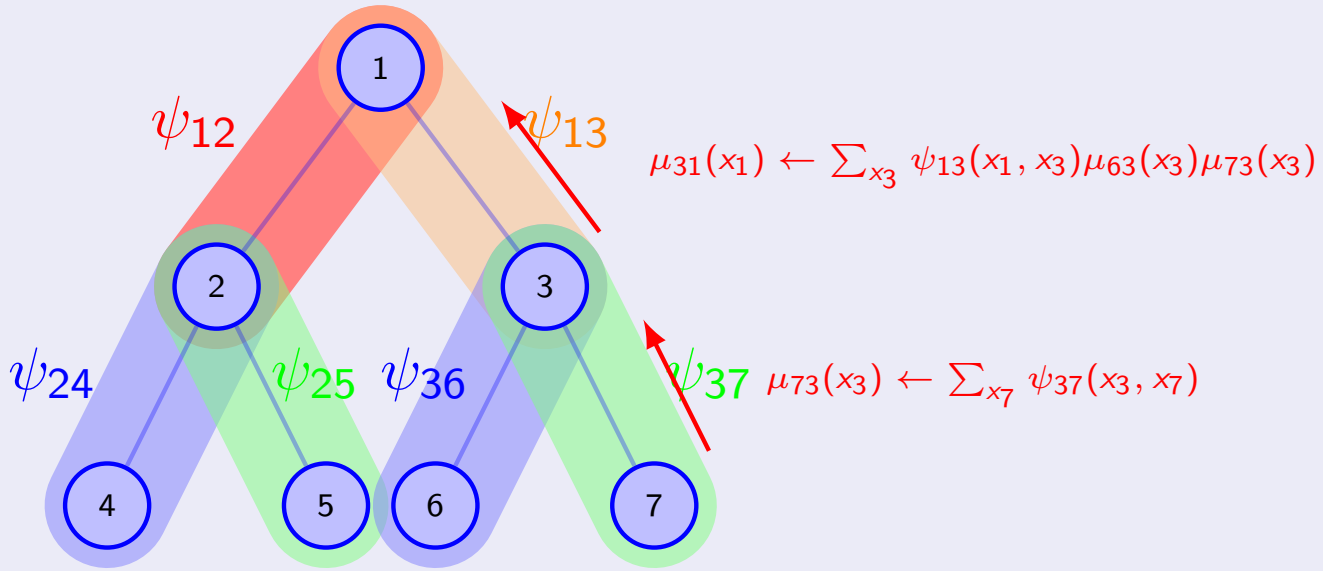
Message passing algorithms are powerful procedures
for exact and approximate inference in graphical models:

- They are derived from variable elimination algorithms
- They can be more powerful for exact inference:
  $\rightarrow$ Using twice as much work, they provide all marginals of a SGM, over all variables.
- They give either exact or approximate solutions:

Exact Message passing on a tree-graphical model $\sim$ variable elimination
Exact Message passing on a tree-factor graph graphical model
Approx Message passing in the general case: (generalized) loopy Belief Propagation (LBP)

Message passing on a tree

Example: Pairwise graphical model

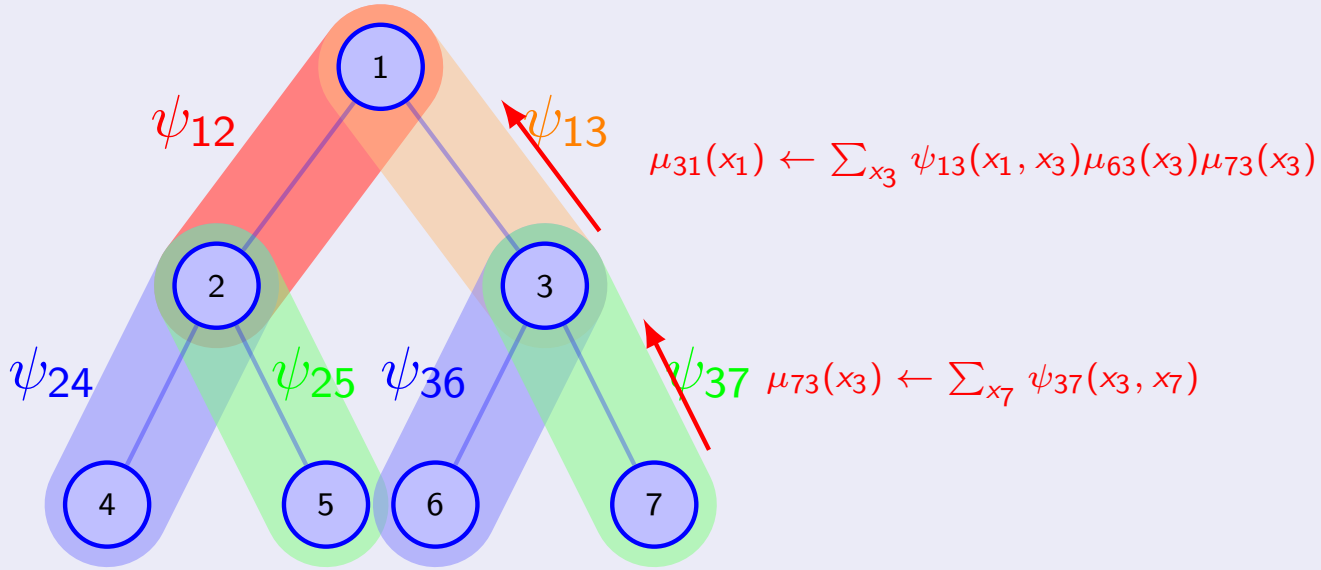


- Messages computation mimics variable elimination
- For marginalization, message updates take the form:

$$\forall x_j, \mu_{ij}(x_j) \leftarrow \frac{1}{K} \sum_{x_i} \psi_{ij}(x_i, x_j)\psi_i(x_i) \prod_{k \neq j,(i,j) \in E} \mu_{ki}(xi)$$

Message passing on a tree

Example: Pairwise graphical model



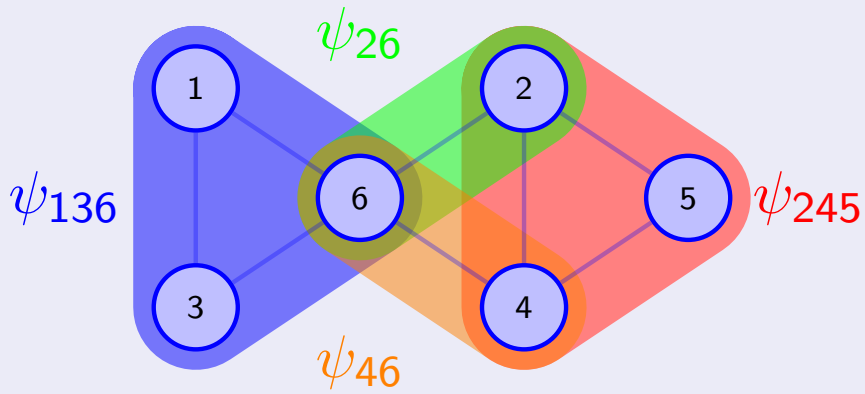
- Once messages have been asynchronously computed along all edges in the two directions, marginal probabilities can be obtained:

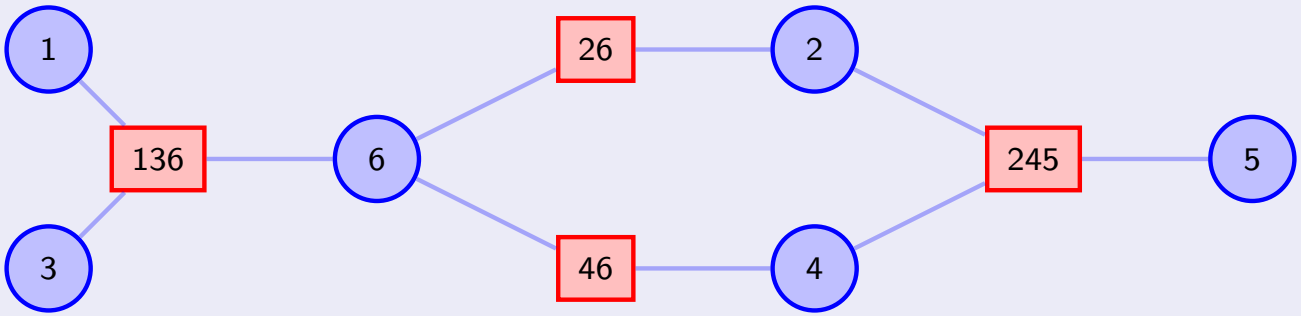
$$p_i(x_i) \leftarrow \frac{1}{K_i}\psi_i(x_i) \prod_{j,(j,i)\in E} \mu_{ji}(x_i), \forall x_i,$$

$$p_{ij}(x_i, x_j) \leftarrow \frac{1}{K_{ij}}\psi_{ij}(x_{ij}) \prod_{k\neq j,(k,i)\in E} \mu_{ki}(x_i) \prod_{l\neq i,(l,j)\in E} \mu_{lj}(x_j).$$

Message passing on a tree factor graph

Example: Hypergraph and equivalent tree factor graph

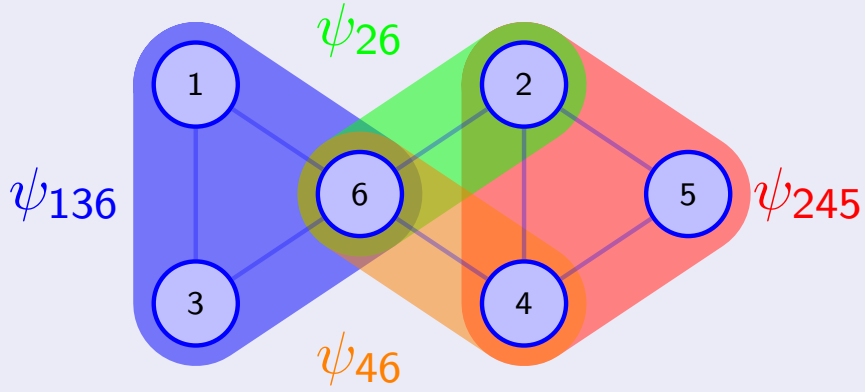


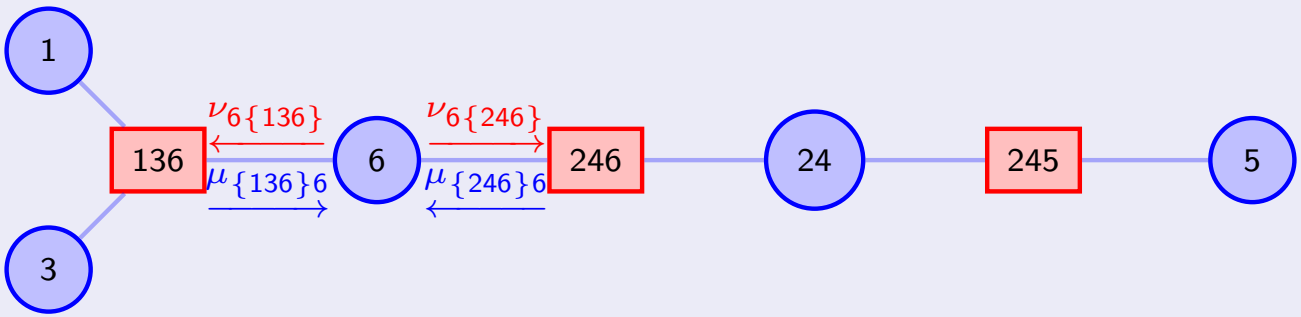


- The factor graph is not a tree...
- But becomes one if we merge variables X_2 and X_4, to get a new variable X'_{24}.

Message passing on a tree factor graph

Example: Hypergraph and equivalent tree factor graph



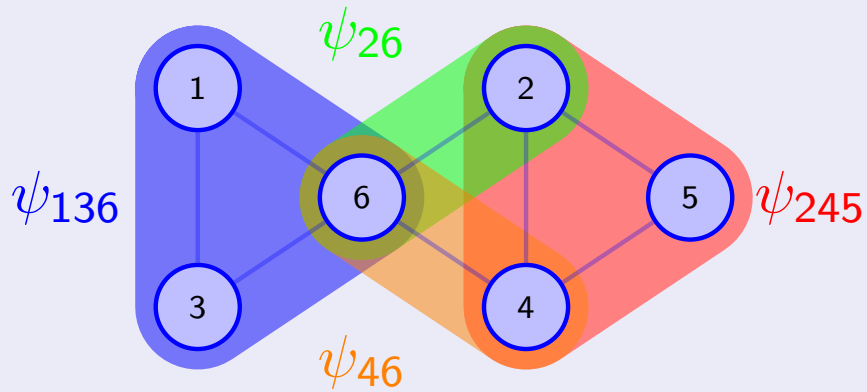


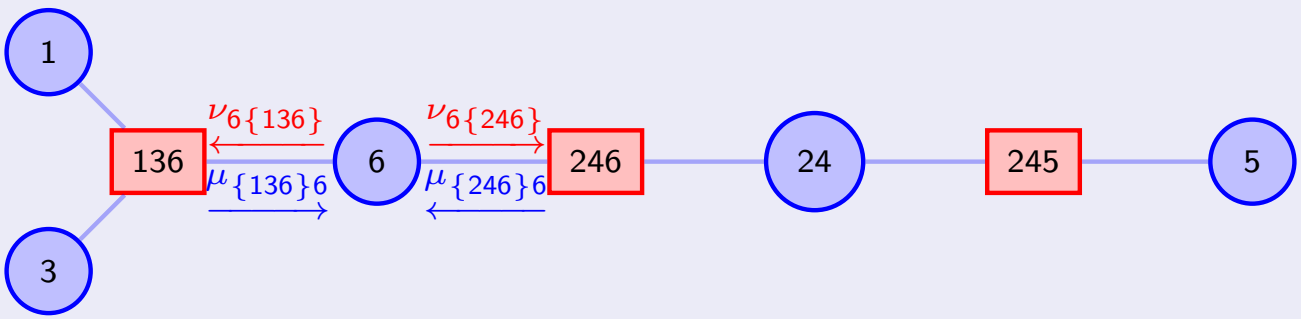
Now that we have a tree, two different kinds of messages are computed:

- Factor-to-variable messages: $\mu_{fi}(x_i)$.
- Variable-to-factor messages: $\nu_{if}(x_i)$.

Message passing on a tree factor graph

Example: Hypergraph and equivalent tree factor graph





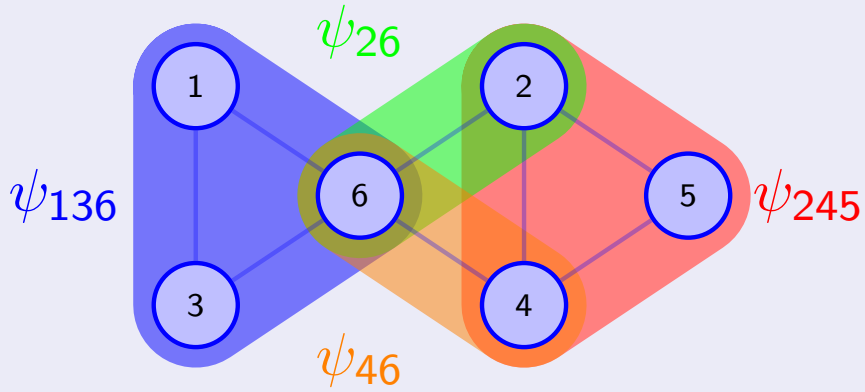
Update rules:

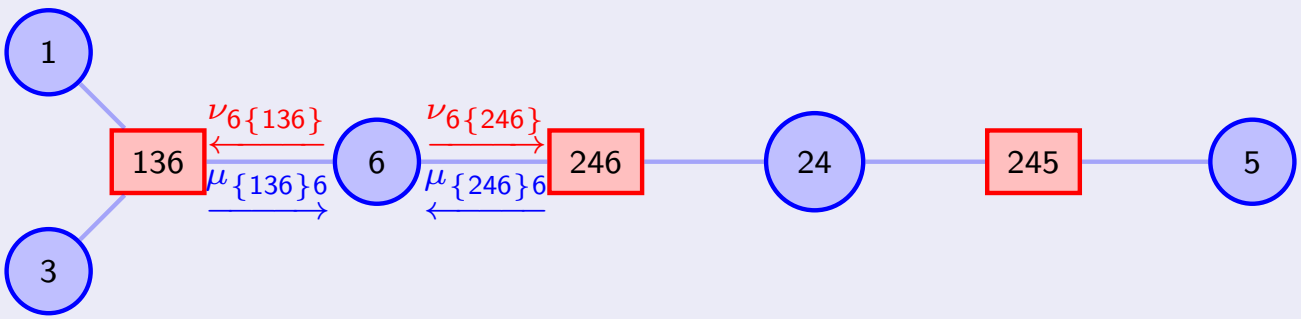
$$\mu_{fi}(x_i) \leftarrow \sum_{x_{f\setminus i}} \left(\psi_f(x_f) \prod_{j \in f\setminus i} \nu_{jf}(x_j) \right), \forall x_i,$$

$$\nu_{if}(x_i) \leftarrow \prod_{f' \neq f, i \in f'} \mu_{f'i}(x_i), \forall x_i.$$

Message passing on a tree factor graph

Example: Hypergraph and equivalent tree factor graph





After two complete passes, marginal probabilities can be computed:

$$p_i(x_i) \leftarrow \frac{1}{K}\psi_i(x_i) \prod_{f, i \in f} \mu_{fi}(x_i), \forall x_i.$$

General case: Loopy Belief Propagation

- When the graphical model is not a tree?
- When the factor graph is not a tree?

$\Rightarrow$ Message passing can still be applied, to approximate marginals!

General case: Loopy Belief Propagation

- When the graphical model is not a tree?
- When the factor graph is not a tree?

$\Rightarrow$ Message passing can still be applied, to approximate marginals!

Loopy Belief Propagation

- Messages sent asynchronously along edges of the graph/factor graph.
- Until an arbitrary convergence condition is met

General case: Loopy Belief Propagation

- When the graphical model is not a tree?
- When the factor graph is not a tree?

$\Rightarrow$ Message passing can still be applied, to approximate marginals!

Loopy Belief Propagation

- Messages sent asynchronously along edges of the graph/factor graph.
- Until an arbitrary convergence condition is met

😢 Convergence to a steady-state is not guaranteed

😢 No guarantee on the "quality" of the approximated marginals

General case: Loopy Belief Propagation

- When the graphical model is not a tree?
- When the factor graph is not a tree?

$\Rightarrow$ Message passing can still be applied, to approximate marginals!

Loopy Belief Propagation

- Messages sent asynchronously along edges of the graph/factor graph.
- Until an arbitrary convergence condition is met

😢 Convergence to a steady-state is not guaranteed

😢 No guarantee on the "quality" of the approximated marginals

😄 LBP steady states have a variational approximation interpretation

😄 "Good" approximation in practice

General case: Generalized Belief Propagation

$\Rightarrow$ Generalized Belief Propagation[1] generalizes Variable elimination and Loopy Belief Propagation

[1] Yedidia, Freeman and Weiss, 2005.

General case: Generalized Belief Propagation

$\Rightarrow$ Generalized Belief Propagation[1] generalizes Variable elimination and Loopy Belief Propagation

Generalized Belief Propagation

- Messages sent asynchronously along edges of a region graph instead of the graph/factor graph.
- A region graph is a directed graph including large and small regions (sets of factors+variables) and satisfy some region graph conditions.
- Different choices of region graph allow different computations:

[1] Yedidia, Freeman and Weiss, 2005.

General case: Generalized Belief Propagation

$\Rightarrow$ Generalized Belief Propagation[1] generalizes Variable elimination and Loopy Belief Propagation

Generalized Belief Propagation

- Messages sent asynchronously along edges of a region graph instead of the graph/factor graph.
- A region graph is a directed graph including large and small regions (sets of factors+variables) and satisfy some region graph conditions.
- Different choices of region graph allow different computations:
  - Junction graph $\rightarrow$ Junction tree method (exact)

[1] Yedidia, Freeman and Weiss, 2005.

General case: Generalized Belief Propagation

$\Rightarrow$ Generalized Belief Propagation[1] generalizes Variable elimination and Loopy Belief Propagation

Generalized Belief Propagation

- Messages sent asynchronously along edges of a region graph instead of the graph/factor graph.
- A region graph is a directed graph including large and small regions (sets of factors+variables) and satisfy some region graph conditions.
- Different choices of region graph allow different computations:
  - Junction graph $\rightarrow$ Junction tree method (exact)
  - Bethe region graph $\rightarrow$ LBP steady states

[1] Yedidia, Freeman and Weiss, 2005.

General case: Generalized Belief Propagation

$\Rightarrow$ Generalized Belief Propagation[1] generalizes Variable elimination and Loopy Belief Propagation

Generalized Belief Propagation

- Messages sent asynchronously along edges of a region graph instead of the graph/factor graph.
- A region graph is a directed graph including large and small regions (sets of factors+variables) and satisfy some region graph conditions.
- Different choices of region graph allow different computations:
  - Junction graph $\rightarrow$ Junction tree method (exact)
  - Bethe region graph $\rightarrow$ LBP steady states
  - Cluster Variational Method region graph $\rightarrow$ Kikuchi approximation of marginals

[1] Yedidia, Freeman and Weiss, 2005.

Summary and concluding remarks

Graphical models

- A convenient approach to model uncertainty and preferences
- A very general model, with variants (combination operator)
- Useful to solve several problems (optimizing, marginalizing...)

A generic algorithmic approach for exact inference (variable elimination) or approximate inference (LBP, GBP)

Summary and concluding remarks

Graphical models

- A convenient approach to model uncertainty and preferences
- A very general model, with variants (combination operator)
- Useful to solve several problems (optimizing, marginalizing...)

A generic algorithmic approach for exact inference (variable elimination) or approximate inference (LBP, GBP)

Other ways to perform inference in graphical models

- Search (maximizing)
- Monte-Carlo simulation (lots of theoretical and practical results)
- Mathematical programming...

Another important question: Infering structure of GM from data!

References

To read more about the topic of this talk:

- N. Peyrard et al. Exact and approximate inference in graphical models: variable elimination and beyond, 2017. https://arxiv.org/pdf/1506.08544.pdf

And some very useful references:

- D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.
- J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

And an home-made Matlab implementation of GBP:

- **GMtoolbox** : `http://www7.inra.fr/mia/T/GMtoolbox/`