



HAL
open science

Exact and approximate inference in graphical models

Régis Sabbadin, Nathalie Peyrard, Marie-Josée Cros, Simon de Givry, Alain Franc, Stephane Robin, Thomas Schiex, Matthieu Vignes

► **To cite this version:**

Régis Sabbadin, Nathalie Peyrard, Marie-Josée Cros, Simon de Givry, Alain Franc, et al.. Exact and approximate inference in graphical models. 10th International Conference on Scalable Uncertainty Management (SUM), Oct 2017, Granada, Spain. hal-02116334

HAL Id: hal-02116334

<https://hal.science/hal-02116334v1>

Submitted on 5 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

SUM'17, Granada, Spain
October, 5, 2017

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 $V = \{1, \dots, n\}$
 and $\odot \in \{\prod, \sum, \min, \max \dots\}$ 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 $V = \{1, \dots, n\}$
and $\odot \in \{\prod, \sum, \min, \max \dots\}$ 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 $V = \{1, \dots, n\}$
and $\odot \in \{\prod, \sum, \min, \max \dots\}$ 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...

- 1 Graphical models
- 2 Inference tasks
- 3 Variable elimination, elimination ordering, treewidth
- 4 Message passing

Definition (Markov chain)

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

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

Definition (Markov chain)

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

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

Definition (Hidden Markov chain)

- $X = (X_1, \dots, X_n)$ is an *unobserved* Markov chain.
- $Z = (Z_1, \dots, Z_n)$ 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)

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

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

Examples of probabilistic graphical models (2)

Definition (Bayesian network)

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

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

Definition (Markov Random Field)

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

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

Definition (Possibilistic networks)

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

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

Definition (Possibilistic networks)

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

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

Definition (Cost Functions networks)

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

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

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

→ We will limit ourselves to single operator graphical models!

Definition (Graphical model)

- Let $X = (X_1, \dots, X_n)$ be a set of variables.
- X_i takes values in $\Lambda_i \subseteq \mathbb{R}$.
- A realization of X is denoted $x = (x_1, \dots, x_n)$, with $x_i \in \Lambda_i$.
- A graphical model over X is a function $\psi : \prod_i \Lambda_i \rightarrow \mathbb{R}$, 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 $V = \{1, \dots, n\}$, $\psi_B : \prod_{i \in B} \Lambda_i \rightarrow \mathbb{R}$ and $\odot \in \{\prod, \sum, \min, \max \dots\}$ is a combination operator.

Why are these models called “graphical”?

They admit “graphical” representations:

- Useful for variables interactions **visualization**
- 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**
⇒ Represent conditional dependence between variables...
- **Undirected graph representation**
⇒ Represent conditional independence between variables...
And useful to compute variable elimination orderings
- **Hypergraph representation**
⇒ Directly represents the graphical model functions scopes
- **Factor graph representation**
⇒ Equivalent to hypergraph representation.
Useful in message passing algorithms

Illustration on a bayesian network

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

Illustration on a bayesian network

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

Directed acyclic graph

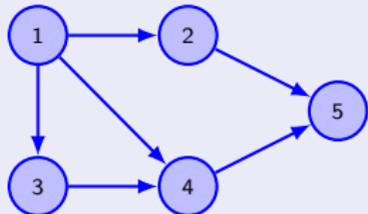
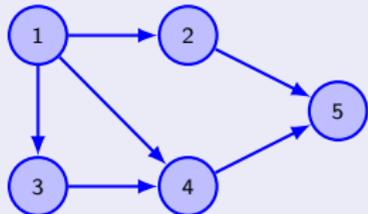


Illustration on a bayesian network

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

Directed acyclic graph



Undirected graph

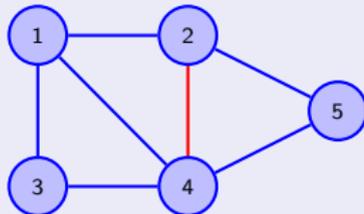


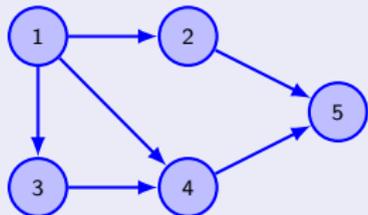
Illustration on a bayesian network

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

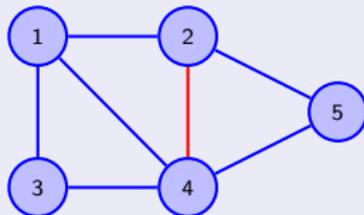
$$\psi_{12}(X_1, X_2) = P(X_1)P(X_2|X_1)$$

$$\psi_{134}(X_1, X_3, X_4) = P(X_3|X_1)P(X_4|X_1, X_3) ; \psi_{245}(X_2, X_4, X_5) = P(X_5|X_2, X_4)$$

Directed acyclic graph



Undirected graph



Hypergraph

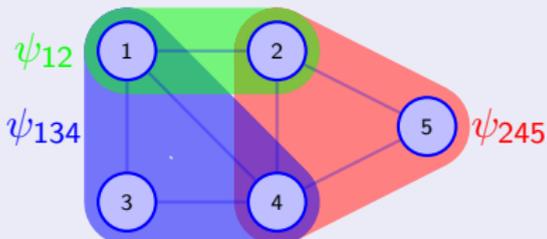


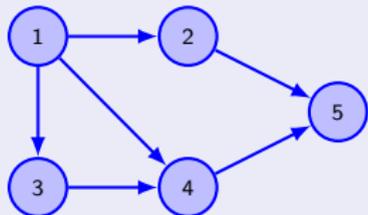
Illustration on a bayesian network

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

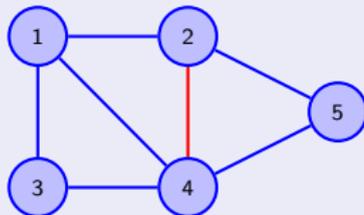
$$\psi_{12}(X_1, X_2) = P(X_1)P(X_2|X_1)$$

$$\psi_{134}(X_1, X_3, X_4) = P(X_3|X_1)P(X_4|X_1, X_3) ; \psi_{245}(X_2, X_4, X_5) = P(X_5|X_2, X_4)$$

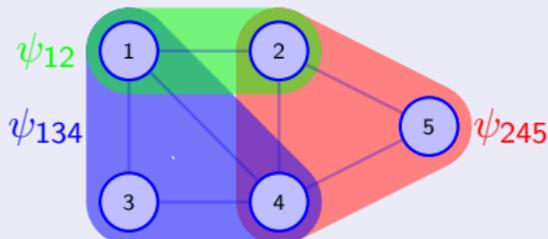
Directed acyclic graph



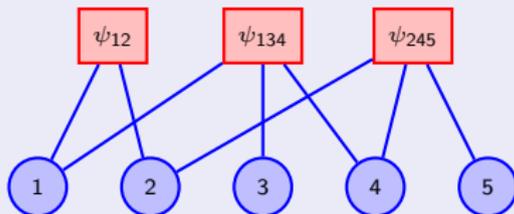
Undirected graph



Hypergraph



Factor graph



- 1 Graphical models
- 2 Inference tasks**
- 3 Variable elimination, elimination ordering, treewidth
- 4 Message passing

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

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

Most likely solutions in possibilistic networks

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

min-cost solutions in cost function networks

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

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

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

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

- 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)

- 1 Graphical models
- 2 Inference tasks
- 3 Variable elimination, elimination ordering, treewidth**
- 4 Message passing

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:

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

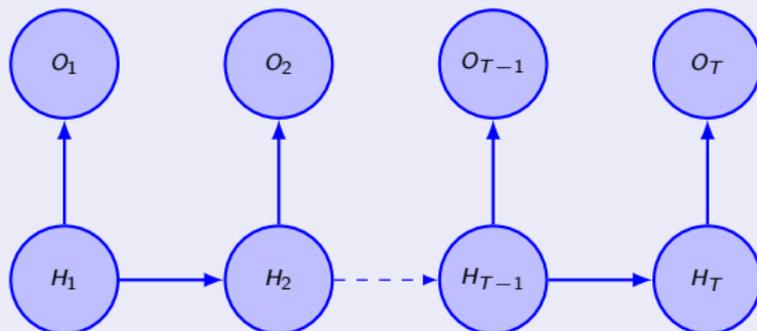
Remarks:

- 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

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

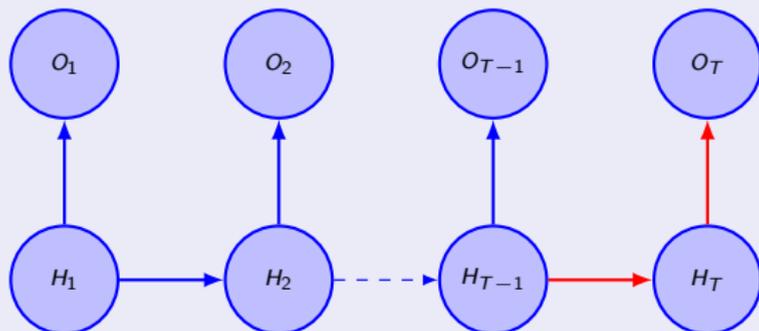
Viterbi algorithm



Variable elimination in HMC: Viterbi algorithm

$$= \max_{h_1, \dots, h_{T-1}} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-2} \psi_{H_i, H_{i+1}}(h_i, h_{i+1}) \psi_{O_i, H_i}(o_i, h_i) \right) \\ \times \psi_{O_{T-1}, H_{T-1}}(o_{T-1}, h_{T-1}) \times \underbrace{\max_{h_T} \psi_{H_{T-1}, H_T}(h_{T-1}, h_T) \psi_{O_T, H_T}(o_T, h_T)}_{\text{New potential function } \psi'_{O_T, H_{T-1}}(o_T, h_{T-1})}$$

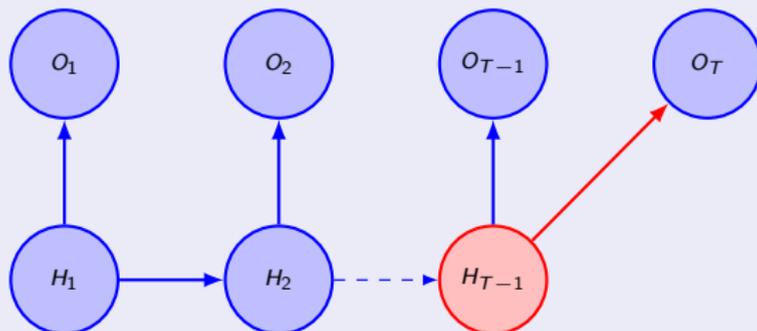
Viterbi algorithm



Variable elimination in HMC: Viterbi algorithm

$$\begin{aligned}
 &= \max_{h_1, \dots, h_{T-1}} \psi_{H_1}(h_1) \left(\prod_{i=1}^{T-2} \psi_{H_i, H_{i+1}}(h_i, h_{i+1}) \psi_{O_i, H_i}(o_i, h_i) \right) \\
 &\times \psi_{O_{T-1}, H_{T-1}}(o_{T-1}, h_{T-1}) \times \underbrace{\max_{h_T} \psi_{H_{T-1}, H_T}(h_{T-1}, h_T) \psi_{O_T, H_T}(o_T, h_T)}_{\text{New potential function } \psi'_{O_T, H_{T-1}}(o_T, h_{T-1})}
 \end{aligned}$$

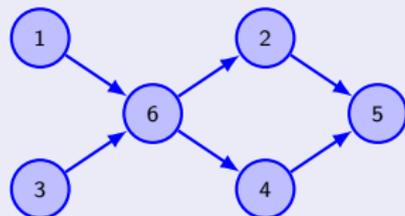
Viterbi algorithm



Variable elimination, bayesian network

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

Directed graph



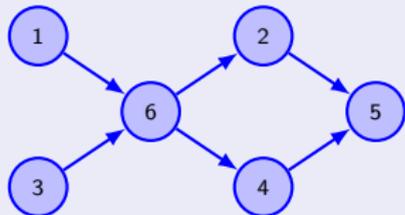
Variable elimination, bayesian network

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

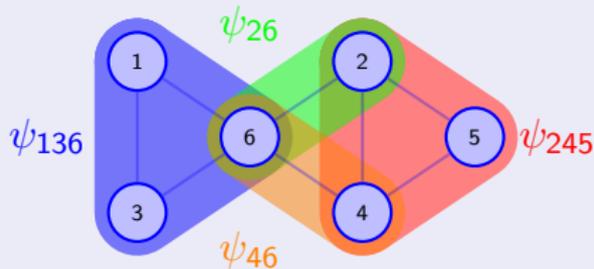
$$P(X_1, \dots, X_6) = \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5)$$

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

Directed graph



Undirected graph / Hypergraph



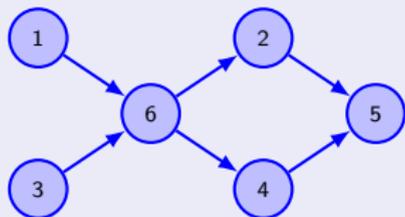
Variable elimination, bayesian network

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

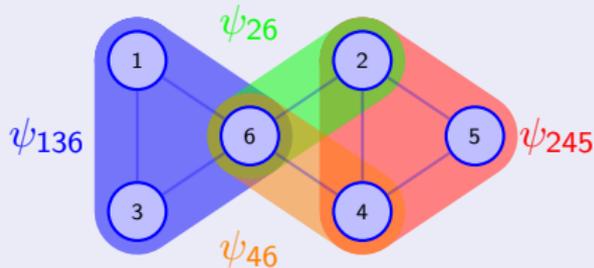
$$P(X_1, \dots, X_6) = \psi_{136}(X_1, X_3, X_6)\psi_{26}(X_2, X_6)\psi_{46}(X_4, X_6)\psi_{245}(X_2, X_4, X_5)$$

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

Directed graph



Undirected graph / Hypergraph



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

Elimination ordering

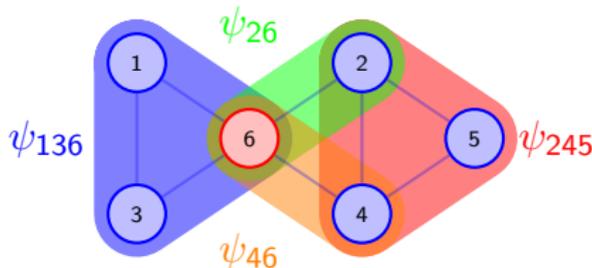
- Variable elimination, for elimination order

$$\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}:$$

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

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

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



Elimination ordering

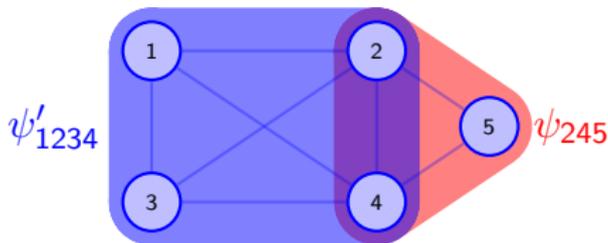
- Variable elimination, for elimination order

$$\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}:$$

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

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

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



Elimination ordering

- Variable elimination, for elimination order

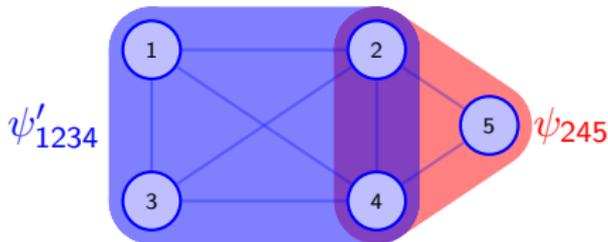
$$\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}:$$

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

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

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

- Is there an elimination order generating “smaller scopes” functions?



Elimination ordering

- Variable elimination, for elimination order

$$\pi = \{X_6, X_5, X_4, X_3, X_2, X_1\}:$$

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

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

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

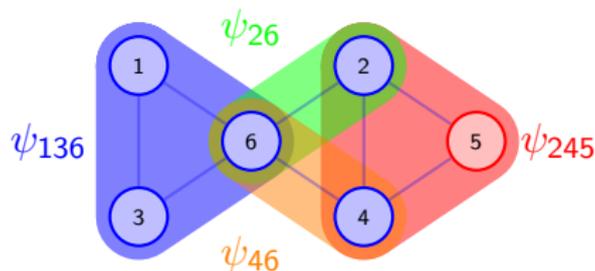
- Is there an elimination order generating “smaller scopes” functions?

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

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

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

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

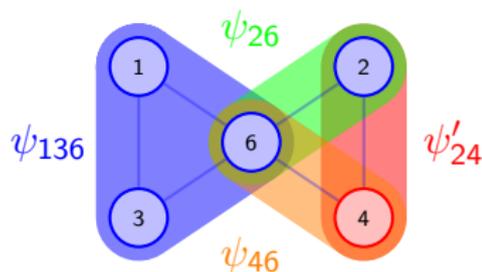


- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

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

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

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

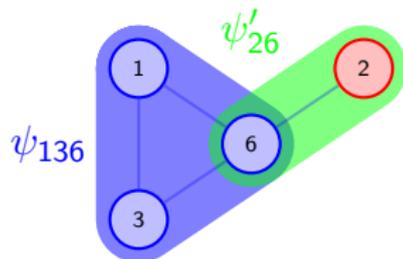


- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

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

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

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

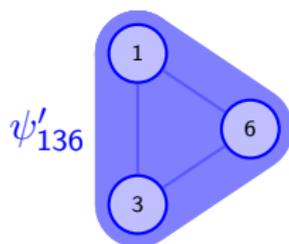


- Yes! $\pi^* = \{X_5, X_4, X_2, X_6, X_3, X_1\}$:

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

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

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



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

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

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

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

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

Treewidth

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

Definition (Variable/vertex elimination)

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

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

Treewidth

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

Definition (Variable/vertex elimination)

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

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

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 $\mathcal{H}^n = \emptyset$).
- ⇒ The induced width is the size of the largest generated hyperedge (minus 1), during the process :

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

Treewidth

Let $\mathcal{H} = \{H_1, \dots, H_m\}$ 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

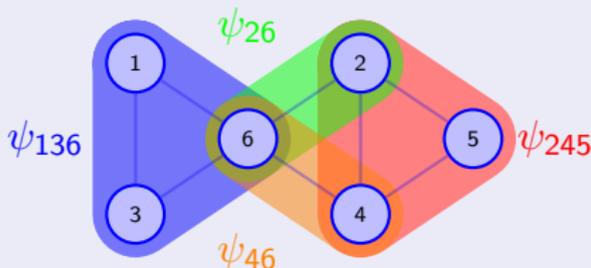
Let $\mathcal{H} = \{H_1, \dots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

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

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

Undirected graph / Hypergraph



Treewidth

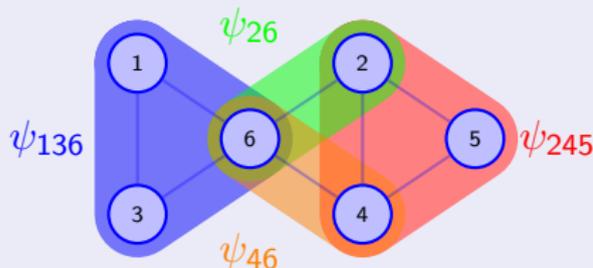
Let $\mathcal{H} = \{H_1, \dots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

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

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

Undirected graph / Hypergraph



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

Treewidth

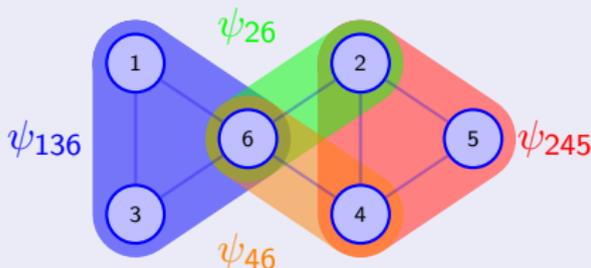
Let $\mathcal{H} = \{H_1, \dots, H_m\}$ be the hypergraph of the graphical model.

Definition (Treewidth)

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

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

Undirected graph / Hypergraph



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

Variable elimination takes exponential space/time in induced width

⇒ Important to find a “good” elimination order (small induced width)

Variable elimination takes exponential space/time in induced width

⇒ 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?

⇒ Not known whether in P or NP-hard

Treewidth computation

Variable elimination takes exponential space/time in induced width

⇒ 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?

⇒ 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

⇒ 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?

⇒ 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)

⇒ In practice, elimination order heuristics (e.g. min-fill).

- 1 Graphical models
- 2 Inference tasks
- 3 Variable elimination, elimination ordering, treewidth
- 4 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:
 - Using twice as much work, they provide **all marginals** of a SGM, over all variables.
- They give either **exact** or **approximate** solutions:

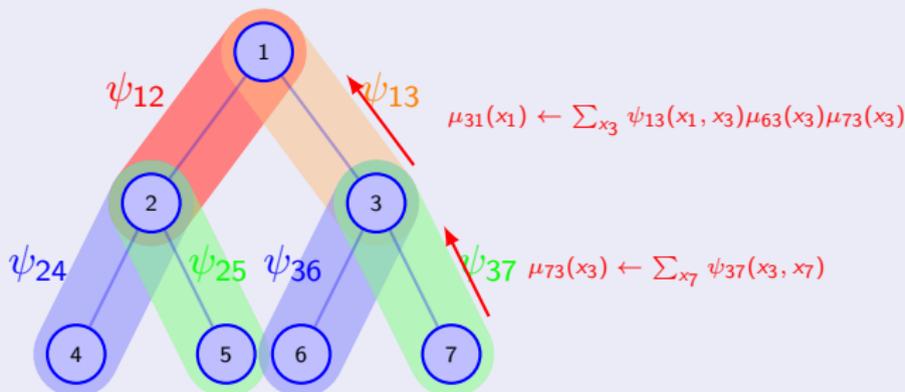
Exact Message passing on a tree-graphical model \sim variable elimination

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

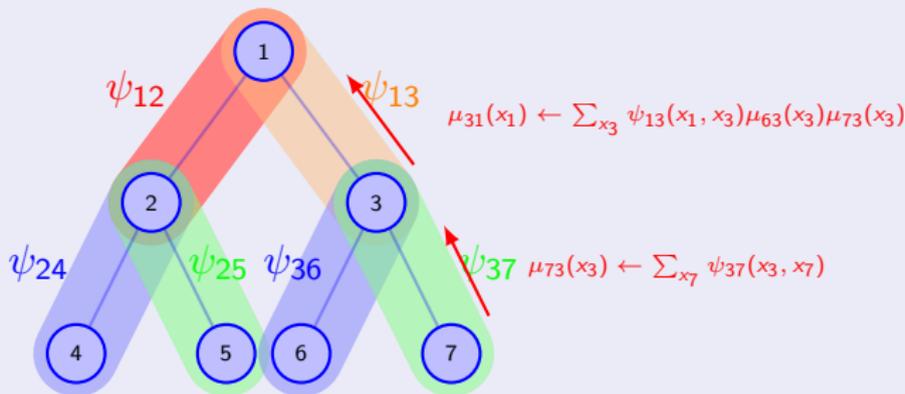


- Messages computation mimics variable elimination
- For marginalization, message updates take the form:

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

Message passing on a tree

Example: Pairwise graphical model



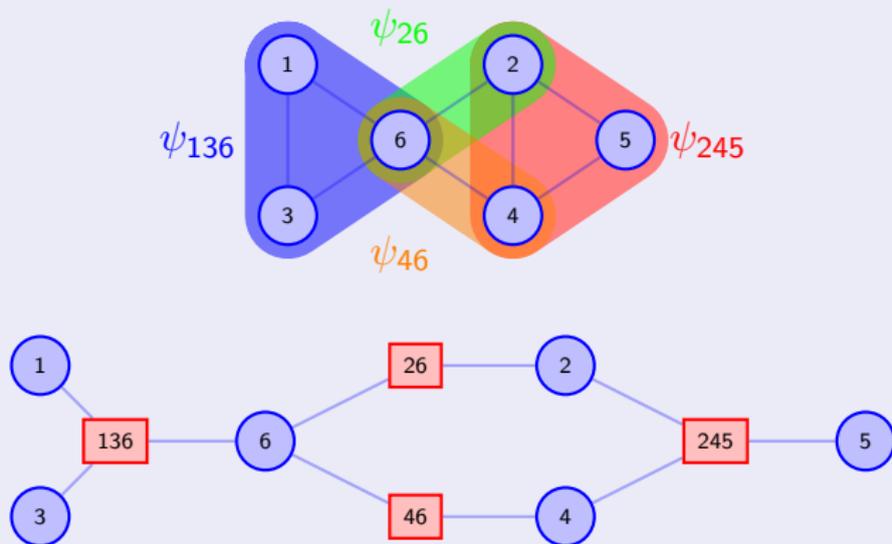
- Once messages have been asynchronously computed along all edges in the two directions, marginal probabilities can be obtained:

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

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

Message passing on a tree factor graph

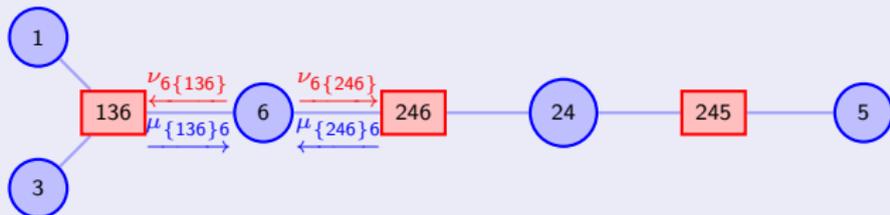
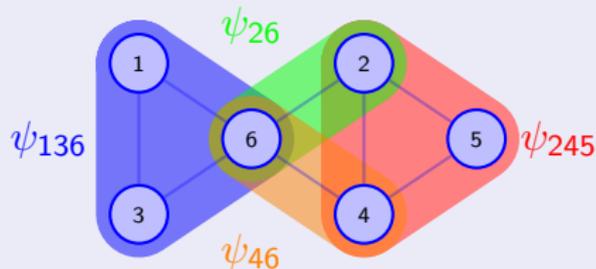
Example: Hypergraph and equivalent tree factor graph



- 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

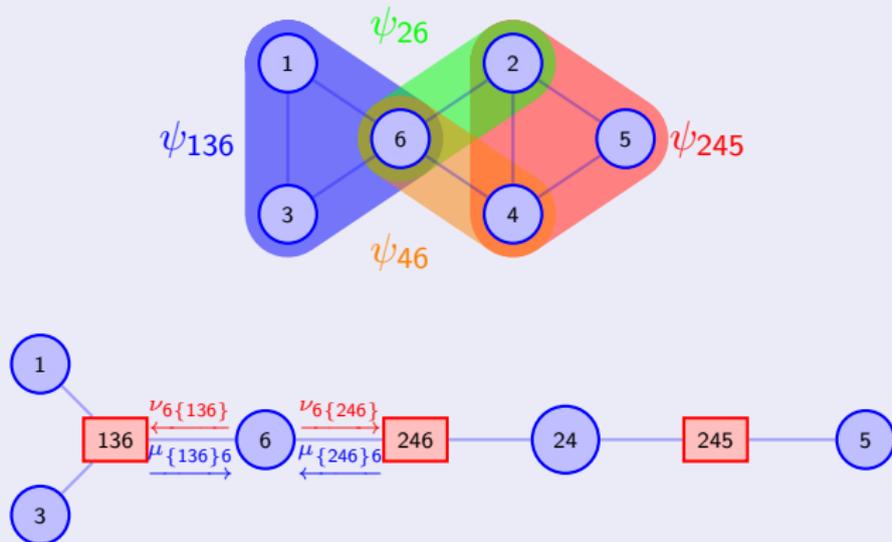


Now that we have a tree, two different kinds of messages are computed:

- 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

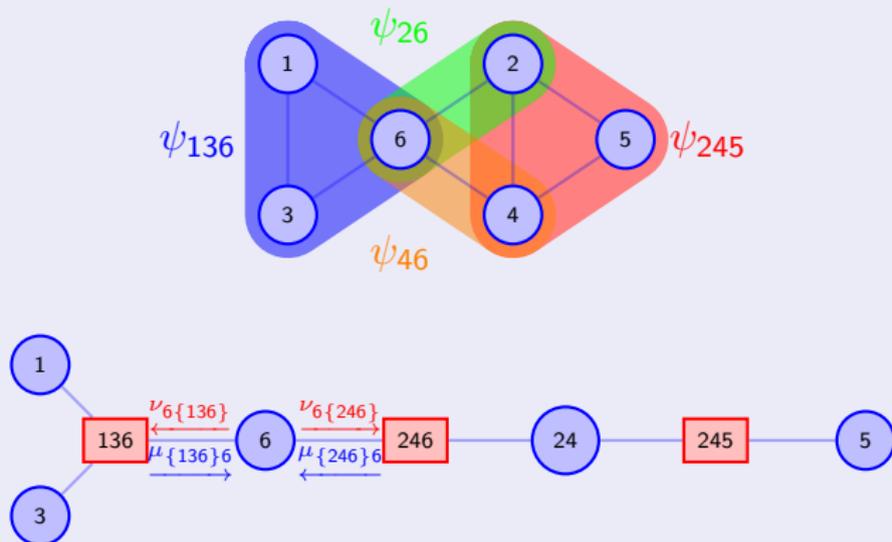


$$\text{Update rules: } \mu_{fi}(x_i) \leftarrow \sum_{x_{f \setminus i}} \left(\psi_f(x_f) \prod_{j \in f \setminus i} \nu_{jf}(x_j) \right), \forall x_i,$$

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

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?

⇒ **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?

⇒ **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?

⇒ **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?

⇒ **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

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)

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)

Other ways to perform inference in graphical models

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

Another important question: **Inferring structure of GM from data!**

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.