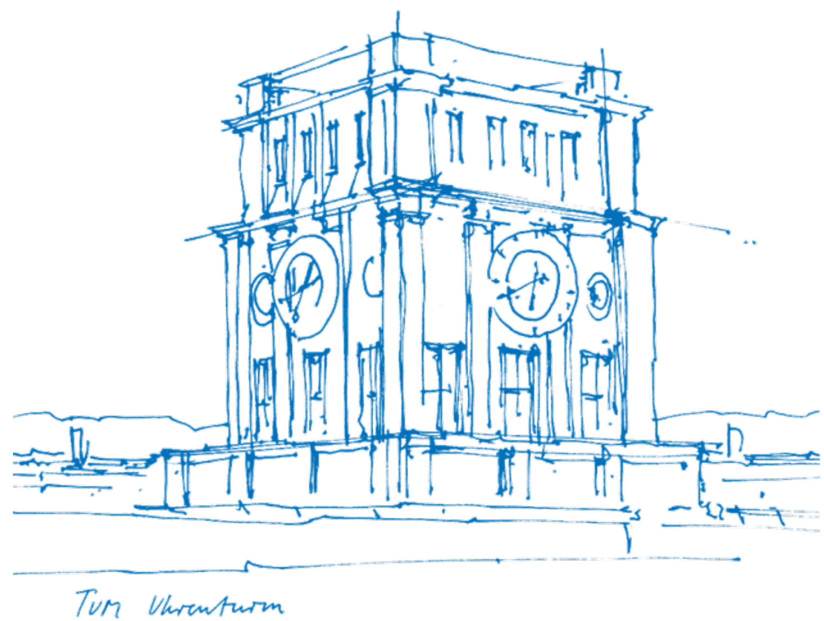
Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

III : Inference on Graphical Models

Tao Wu, Yuesong Shen, Zhenzhang Ye

Computer Vision & Artificial Intelligence
Technical University of Munich



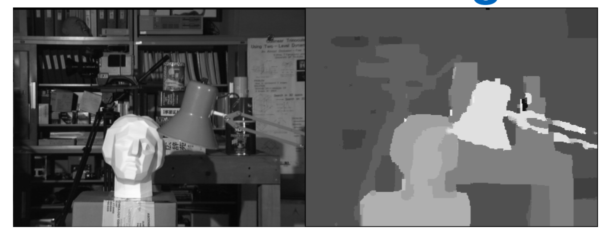
Motivation

– Many computer vision tasks boil down to inference on graphical models.

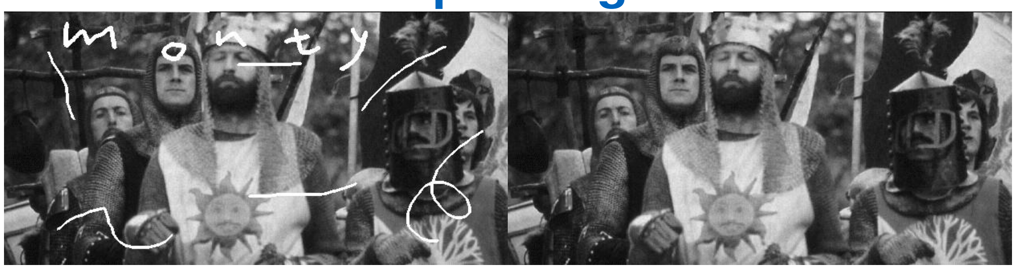
Denoising

Optical flow

Stereo matching



Inpainting



Super-resolution

1. **Probabilistic inference**: compute marginal distribution

$$p(y) = \sum_x p(y, x).$$

2. **MAP inference**: compute maximum of conditional distribution

$$\arg\max_y p(y|x).$$

Exact Inference

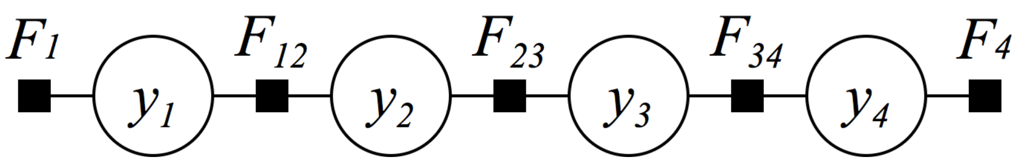
Outline of the Section

- Basic idea: Variable elimination.
- Junction tree algorithm on arbitrary MRFs.
- Belief propagation on tree factor graphs.

Example: Marginal Query on a "Chain" MRF

Joint distribution represented by MRF:

$$p(y_1, y_2, y_3, y_4) = \frac{1}{Z}\phi_1(y_1) \cdot \phi_{12}(y_1, y_2) \cdot \phi_{23}(y_2, y_3) \cdot \phi_{34}(y_3, y_4) \cdot \phi_4(y_4).$$



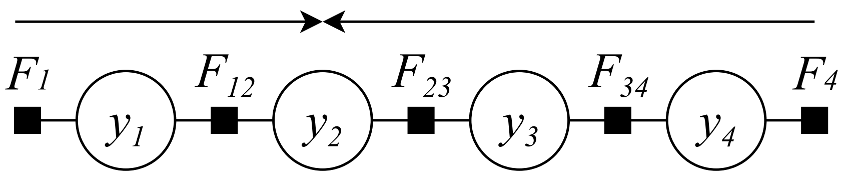
Query about marginal distribution $p(y_2) = ?$

Variable Elimination

Apply **variable elimination** (VE) to the marginal query:

$$
\begin{aligned}
p(y_2) &= \sum_{y_1,y_3,y_4} p(y_1, y_2, y_3, y_4) \\
&= \sum_{y_1,y_3,y_4} \frac{1}{Z}\phi_1(y_1)\phi_{12}(y_1, y_2)\phi_{23}(y_2, y_3)\phi_{34}(y_3, y_4)\phi_4(y_4) \\
&= \frac{1}{Z}\underbrace{\sum_{y_1}\Big(\phi_1(y_1)\phi_{12}(y_1, y_2)\Big)}_{=:\, m_{1\to 2}(y_2)}\sum_{y_3}\Big(\phi_{23}(y_2, y_3)\underbrace{\sum_{y_4}\big(\phi_{34}(y_3, y_4)\phi_4(y_4)\big)}_{=:\, m_{4\to 3}(y_3)}\Big) \\
&= \frac{1}{Z}m_{1\to 2}(y_2)\underbrace{\sum_{y_3}\Big(\phi_{23}(y_2, y_3)m_{4\to 3}(y_3)\Big)}_{=:\, m_{3\to 2}(y_2)} \\
&= \frac{1}{Z}m_{1\to 2}(y_2)m_{3\to 2}(y_2), \\
Z &= \sum_{y_2} m_{1\to 2}(y_2)m_{3\to 2}(y_2).
\end{aligned}
$$

Variable Elimination



- This algorithm is called **sum-product** VE.
- Sum-product VE yields *exact* inference (of one node marginal) on any *tree-structured factor graph*.
- A similar algorithm can be derived for MAP inference – simply switch all "sum" to "max". The resulting algorithm is called **max-product** VE.
- We shall consider two different extensions beyond VE:

1. Inference on arbitrary MRFs? $\rightsquigarrow$ **Junction tree algorithm**.
2. Compute all node/factor marginals at one shot? $\rightsquigarrow$ **Belief propagation**.

Junction Tree

- For an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the **junction tree** of $\mathcal{G}$ is a tree $\mathcal{T}$ s.t.
  1. The nodes of $\mathcal{T}$ consist of the *maximal cliques* of $\mathcal{G}$.
  2. The edge S_{ij} between two nodes C_i, C_j of $\mathcal{T}$ (i.e. two maximal cliques of $\mathcal{G}$) is given by $S_{ij} = C_i \cap C_j$ (known as the *running intersection property*).
- A graph is **triangulated** if every cycle of length ≥ 4 has a *chord*. (A chord is an edge that is not part of the cycle but connects two vertices of the cycle.)
- Theorem [Lauritzen '96]: A graph has a junction tree iff it is triangulated.

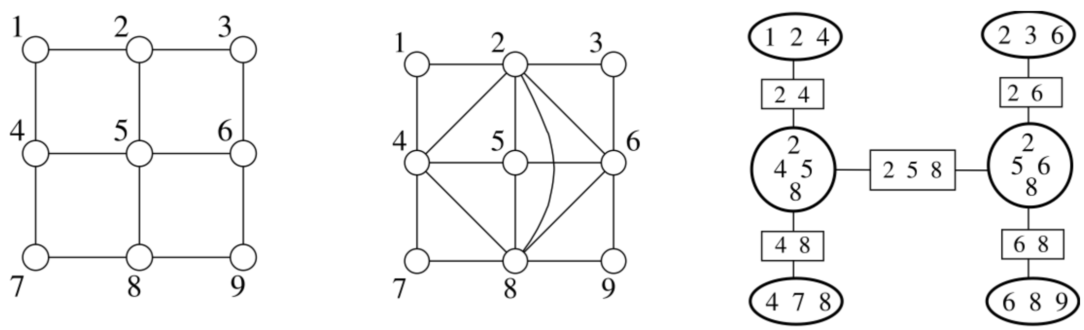
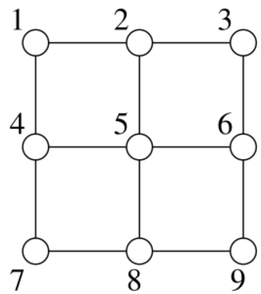
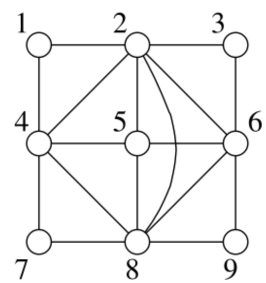


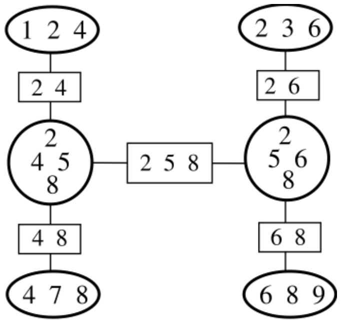
Figure:[1] (a) Original graph; (b) Triangulated graph; (c) Junction tree for the graph in (b).

[1] Wainwright and Jordan, "Graphical Models, Exponential Families, and Variational Inference".

Junction Tree Algorithm







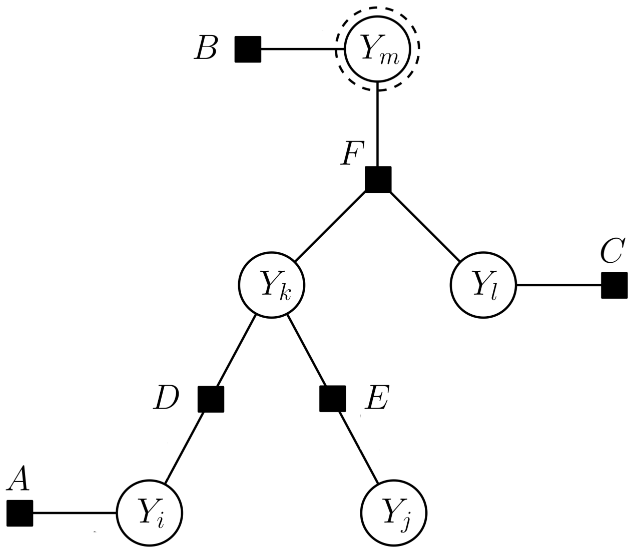
Sum-product message passing on a junction tree $\mathcal{T}$ appears like:

$$m_{C_i \to C_j}(y_{C_j \setminus C_i}) = \sum_{y_{C_i \setminus C_j}} \psi_{C_i}(y_{C_i}) \prod_{C_k \in \mathcal{N}_{\mathcal{T}}(C_i) \setminus \{C_j\}} m_{C_k \to C_i}(y_{C_i \setminus C_k}).$$

Overall **junction tree algorithm** for exact inference on an arbitrary MRF:

1. Given an MRF with cycles, triangulate it by adding edges as necessary.
2. Form a junction tree $\mathcal{T}$ for the triangulated MRF.
3. Run VE on the junction tree $\mathcal{T}$.

Belief Propagation on Tree Factor Graphs[2]



- Factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$: assumed to be a tree.
- Neighbors of a variable or factor node:

$$N_{\mathcal{G}}(i) = \{F \in \mathcal{F} : (i, F) \in \mathcal{E}\},$$
$$N_{\mathcal{G}}(F) = \{i \in \mathcal{V} : (i, F) \in \mathcal{E}\}.$$

- (Log-domain) energies: $E_F(y_F) = -\log \phi_F(y_F)$.

[2] Illustrations for BP are extracted from [Nowozin/Lampert '11].

BP: Leaf-to-Root Stage

0. Pick $Y_r \in \mathcal{V}$ as the tree root (e.g. Y_m in the figure).

1a. Schedule the leaf-to-root messages.

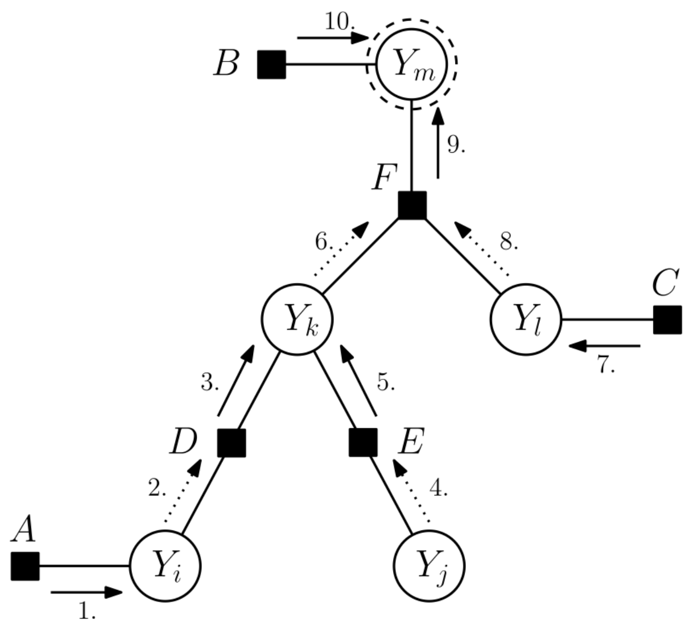


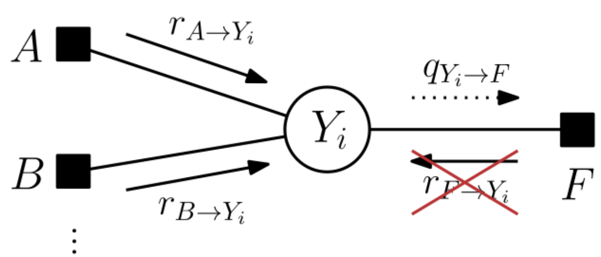
Figure: Belief propagation: leaf-to-root stage.

1b. Compute all leaf-to-root messages (detailed in the next slide).

BP: Compute Messages

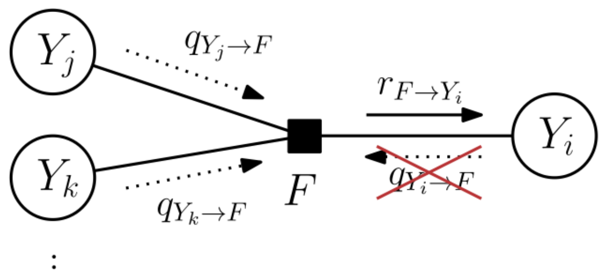
- Compute variable-to-factor message:

$$q_{i\to F}(y_i) = \sum_{F' \in \mathcal{N}_{\mathcal{G}}(i)\setminus\{F\}} r_{F'\to i}(y_i).$$



- Compute factor-to-variable message:

$$r_{F\to i}(y_i) = \log \sum_{y_{F\setminus\{i\}}} \exp\Big(-E_F(y_F) + \sum_{i' \in \mathcal{N}_{\mathcal{G}}(F)\setminus\{i\}} q_{i'\to F}(y_F)\Big).$$



BP: Compute the Partition Function

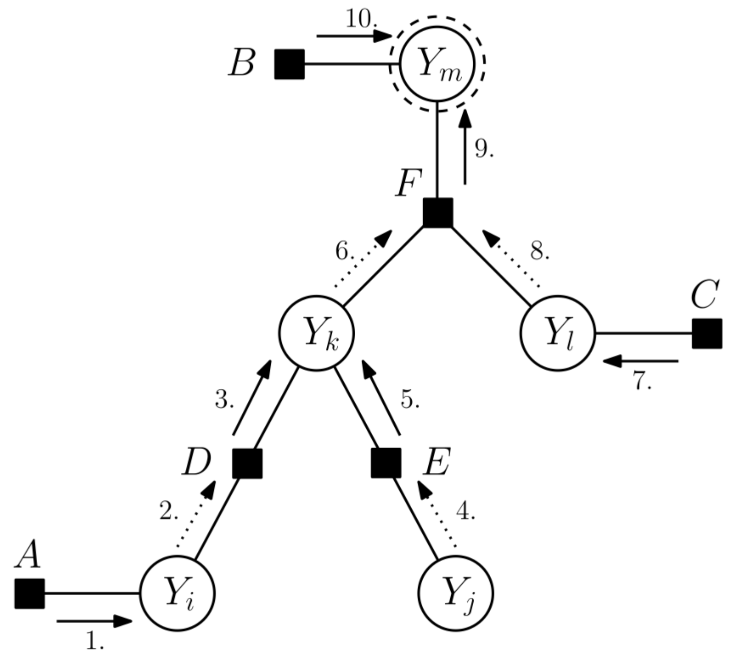


Figure: Belief propagation: leaf-to-root stage.

1c. Compute the log partition function:

$$\log Z = \log \sum_{y_r} \exp\left(\sum_{F \in \mathcal{N}_{\mathcal{G}}(r)} r_{F \to r}(y_r) \right).$$

BP: Root-to-Leaf Stage

2a. Schedule the root-to-leaf messages.

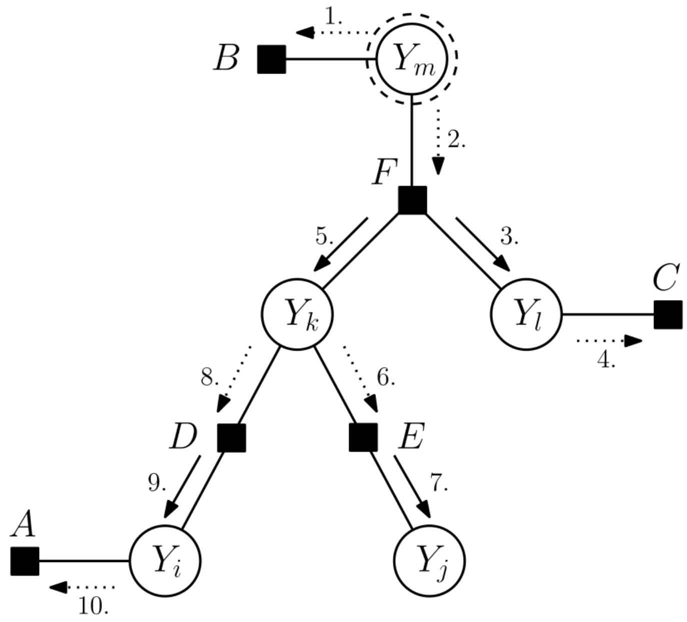


Figure: Belief propagation: root-to-leaf stage.

2b. Compute the root-to-leaf messages using the same formulas on page 12.

BP: Compute Factor / Variable Marginals

2c. Alongside Step 2b, combine messages and compute factor marginals:

$$\mu_F(y_F) := p(y_F) = \exp\Big(-E_F(y_F) + \sum_{i \in \mathcal{N}_\mathcal{G}(F)} q_{i \to F}(y_i) - \log Z \Big),$$

as well as variable marginals:

$$\mu_i(y_i) := p(y_i) = \exp\Big(\sum_{F \in \mathcal{N}_\mathcal{G}(i)} r_{F \to i}(y_i) - \log Z \Big).$$

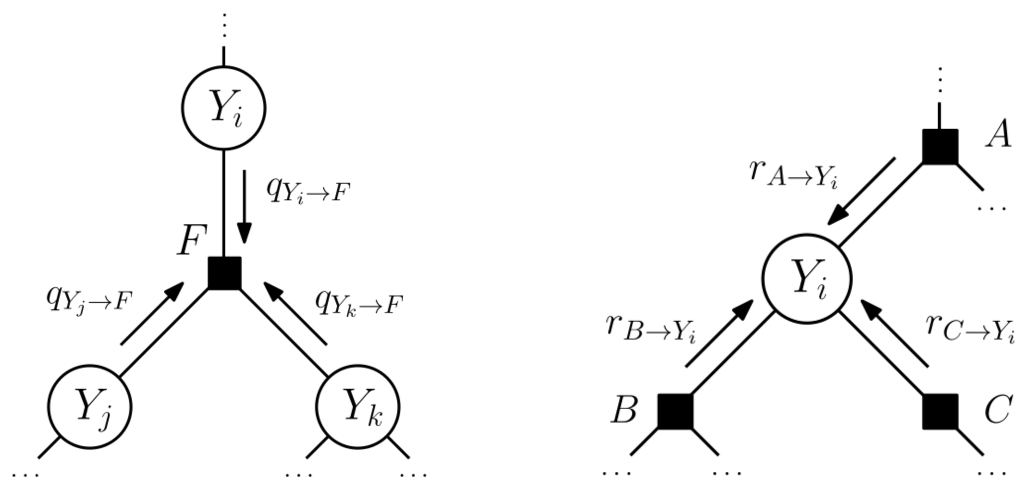


Figure: (left) Factor marginal; (right) Variable marginal.

BP on Pairwise MRFs

On a pairwise MRF $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, the joint distribution is factorized by

$$p(y) = \exp\Big(- \sum_{i \in \mathcal{V}} E_i(y_i) - \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j) - \log Z \Big).$$

BP on such pairwise MRF can be simplified:

- Node-to-node message is computed by

$$m_{i \to j}(y_j) = \log \sum_{y_i} \exp\Big(- E_i(y_i) - E_{ij}(y_i, y_j) + \sum_{k \in \mathcal{N}_{\mathcal{H}}(i) \setminus \{j\}} m_{k \to i}(y_i) \Big).$$

- Node marginal is computed by

$$\mu_i(y_i) = \exp\Big(- E_i(y_i) + \sum_{k \in \mathcal{N}_{\mathcal{H}}(i)} m_{k \to i}(y_i) - \log Z \Big).$$

Further Reading

- Koller & Friedman, Chapters 9, 10.
- Murphy, Chapter 20.
- Nowozin & Lampert, Section 3.1.