Machine Learning 2

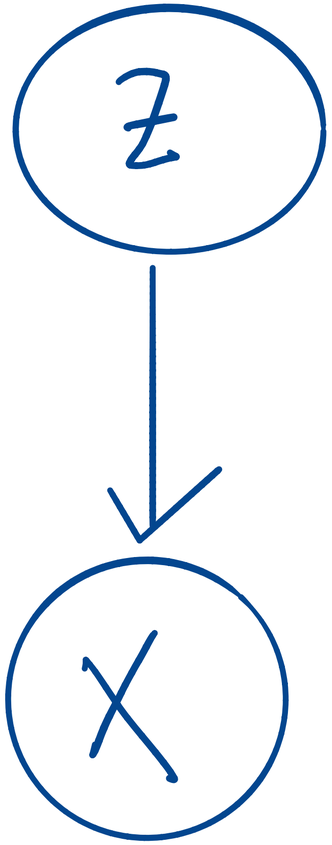
Inference
- Motivation

Patrick Forré

What is Inference?

- Let a joint probability distribution $p(x_1, \ldots, x_M)$ be represented by the computer, e.g. probability tables, factorized forms, via parameters, etc.
- Inference in our context means to derive or compute a representation of certain properties of $p(x_1, \ldots, x_M)$, e.g.
  - marginals distributions: $p(x_k, x_j)$,
  - conditionals distributions: $p(x_k, x_j \mid x_i, x_v)$
  - modes: $x^* \in \arg\max_x p(x)$
  - etc.

Example: Gaussian



$$p(z) = \mathcal{N}(z \mid \mu, \Sigma)$$

$$p(x \mid z) = \mathcal{N}(x \mid Az + b, S)$$

$$p(x, z) = p(x \mid z) \cdot p(z) \stackrel{?}{=} \mathcal{N}\left(\begin{pmatrix} z \\ x \end{pmatrix} \mid m, T\right)$$

joint

$$m = \begin{pmatrix} \mu \\ A\mu + b \end{pmatrix}, \quad T = \begin{pmatrix} \Sigma & \Sigma A^T \\ A\Sigma & S + A\Sigma A^T \end{pmatrix}$$

$$p(x) = \mathcal{N}(x \mid A\mu + b, S + A\Sigma A^T)$$

marginal

Example: Gaussian

$$\mathcal{N}\left(\begin{pmatrix} z \\ x \end{pmatrix} \middle| \begin{pmatrix} \mu \\ A\mu + b \end{pmatrix}, \begin{pmatrix} \Sigma & \Sigma A^T \\ A\Sigma & A\Sigma A^T + S \end{pmatrix}\right)$$

joint

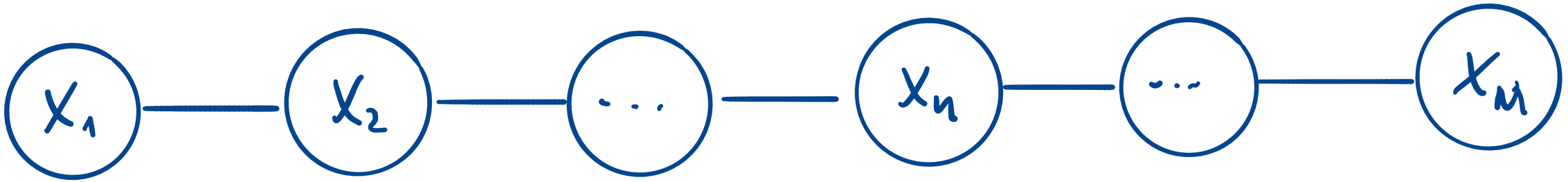
conditional

$$p(z|x) = \mathcal{N}\left(z \middle| \mu + \Sigma A^T (A\Sigma A^T + S)^{-1}(x - A\mu - b),\ \Sigma - \Sigma A^T (A\Sigma A^T + S)^{-1} A\Sigma\right)$$

Inverse $\sim O(n^3)$

(might be infeasible)

Example: Discrete Markov Chain



$$p(x) = \frac{1}{Z} \cdot \prod_{i=1}^{M-1} \psi_{\{i,i+1\}}(x_i, x_{i+1})$$

each x_i with K classes

$$p(x_n) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \sum_{x_M} p(x_1, \ldots x_{n-1}, x_n, x_{n+1}, \ldots x_M) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_M} \frac{1}{Z} \prod_i \psi_{\{i,i+1\}}(x_i, x_{i+1})$$

$x_1, x_2, \ldots x_{n-1}, x_{n+1}, x_M$

K^{M-1} states to add up

Example: Discrete Markov Chain

$$p(\mathbf{x}_n) = \sum_{x_1} \cdots \sum_{\hat{x}_n} \cdots \sum_{x_M} \frac{1}{Z} \prod_{i=1}^{M} \psi_{\{i,i+1\}}(x_i, x_{i+1})$$

$$= \frac{1}{Z} \sum_{x_1} \sum_{x_2} \cdots \psi_{\{1,2\}}(x_1, x_2) \cdots \psi_{\{n-1,n\}}(x_{n-1}, \mathbf{x}_n) \sum_{x_{n+1}} \psi_{\{n,n+1\}}(\mathbf{x}_n, x_{n+1}) \cdots \underbrace{\left(\sum_{x_M} \psi_{\{M-1,M\}}(x_{M-1}, x_M) \right)}_{K^2}$$

M times

$$\sim M \cdot K^2 \ll K^{M-1}$$

Different ways to do inference

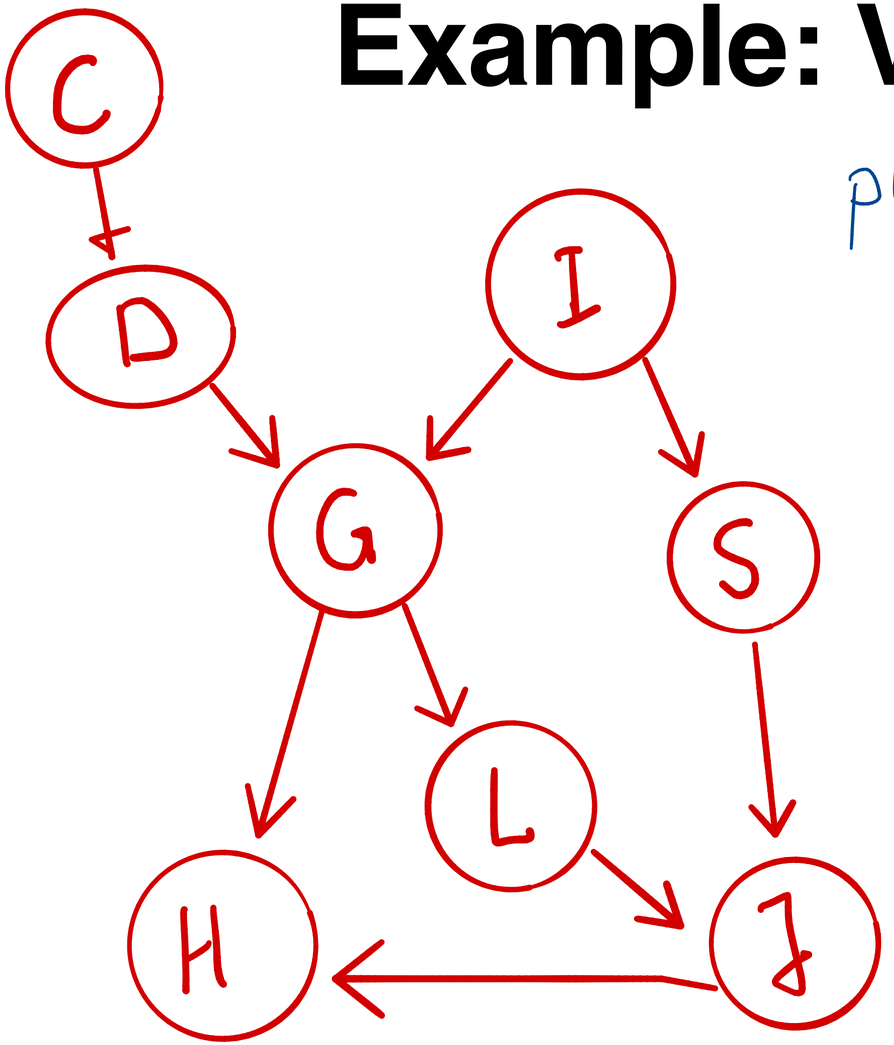
- Exact Inference (desirable, but often computational infeasible)
  - Exact Inference in Graphical Models (exploiting sparse representations)
- Approximate Inference
  - Variational Inference (analytic or numerical approximations)
    - Expectation-Maximization (EM) Algorithm
    - Variational Bayes Mean Field Approximation
    - Variational Autoencoder (VAE)
  - Monte Carlo Sampling Methods (approximation creating and using (weighted) samples)
    - Importance, Slice, Rejection Sampling, etc.
    - Metropolis-Hastings MCMC

Machine Learning 2

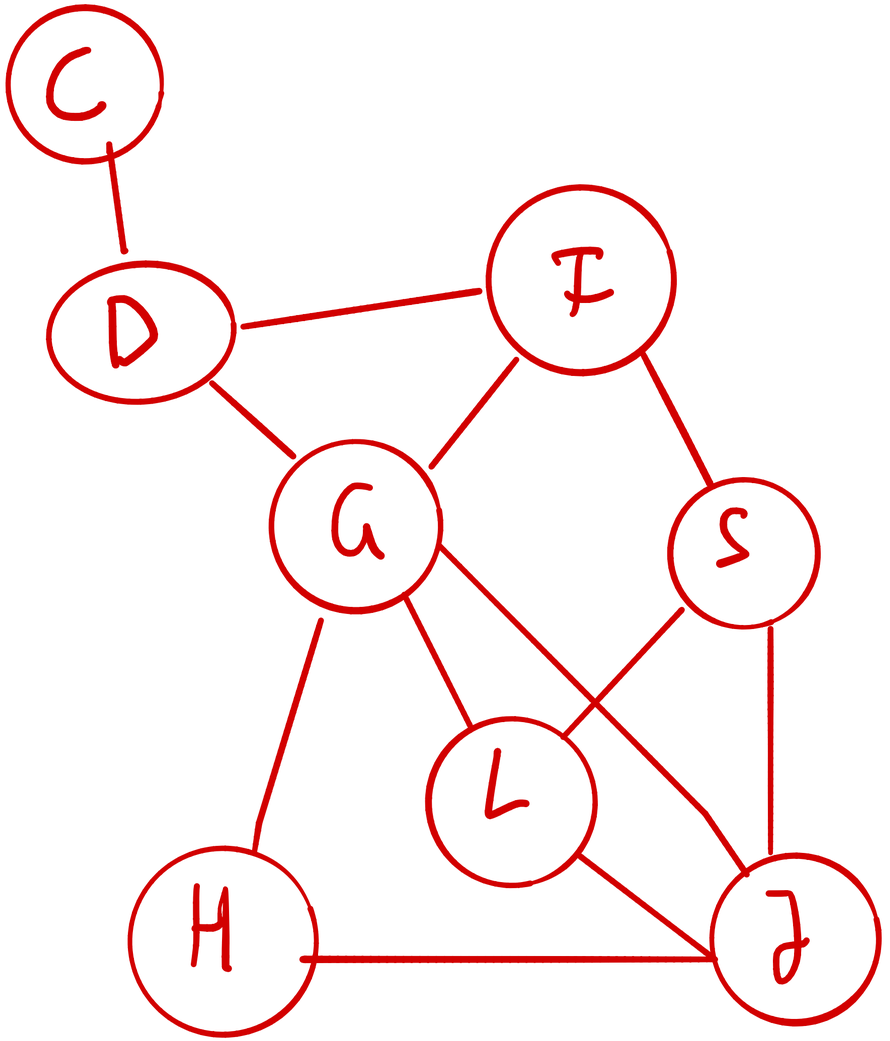
Graphical Models
- Exact Inference
- Variable Elimination Algorithm

Patrick Forré

Example: Variable Elimination



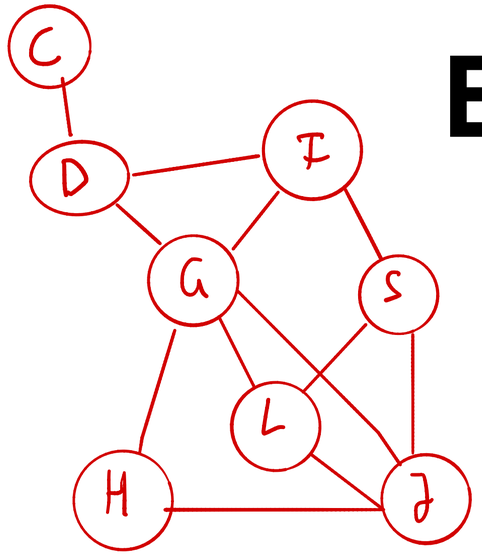
$p(J)$



$p(X)$

$= p(C) \cdot p(D|C)\, p(I) \cdot p(G|I,D) \cdot p(S|I) \cdot p(L|G) \cdot p(J|L,S) \cdot p(H|G,J)$

$= \psi(C)\ \psi(D,C) \cdot \psi(I)\ \psi(G,I,D)\ \psi(S,I)\ \psi(L,G)\ \psi(J,L,S)\ \psi(H,G,J)$



Example: Variable Elimination

$$p(X) = \psi(C)\cdot\psi(D,C)\cdot\psi(I)\cdot\psi(G,I,D)\cdot\psi(S,I)\cdot\psi(L,G)\cdot\psi(J,L,S)\cdot\psi(H,G,J)$$

$$p(J) = \sum_{L}\sum_{S}\sum_{G}\sum_{H}\sum_{I}\sum_{D}\sum_{C} p(X)$$

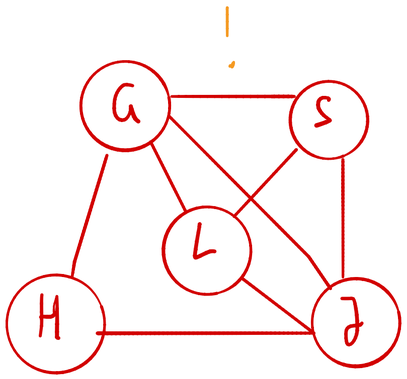
$$= \sum_{L,S}\psi(J,L,S)\cdot\sum_{G}\psi(L,G)\sum_{H}\psi(H,G,J)\cdot\sum_{I}\psi(S,I)\cdot\psi(I)\sum_{D}\psi(G,I,D)\sum_{C}\psi(C)\psi(D,C).$$

$$= \sum_{L,S}\psi(J,L,S)\cdot\sum_{G}\psi(L,G)\sum_{H}\psi(H,G,J)\cdot\sum_{I}\psi(S,I)\cdot\psi(I)\sum_{D}\psi(G,I,D)\cdot\phi_1(D)$$

$$= \sum_{L,S}\psi(J,L,S)\cdot\sum_{G}\psi(L,G)\sum_{H}\psi(H,G,J)\cdot\sum_{I}\psi(S,I)\cdot\psi(I)\cdot\phi_2(G,I)$$

$$= \sum_{L,S}\psi(J,L,S)\cdot\sum_{G}\psi(L,G)\sum_{H}\psi(H,G,J)\cdot\phi_3(S,G)$$

Example: Variable Elimination



$$p(X) = \psi(C) \cdot \psi(D,C) \cdot \psi(I) \cdot \psi(G,I,D) \cdot \psi(S,I) \cdot \psi(L,G) \cdot \psi(J,L,S) \cdot \psi(H,G,J)$$

$$p(J) = \sum_{L} \sum_{S} \sum_{G} \sum_{H} \left(\sum_{I} \sum_{D} \sum_{C} p(X) \right)$$

$$p(J)$$

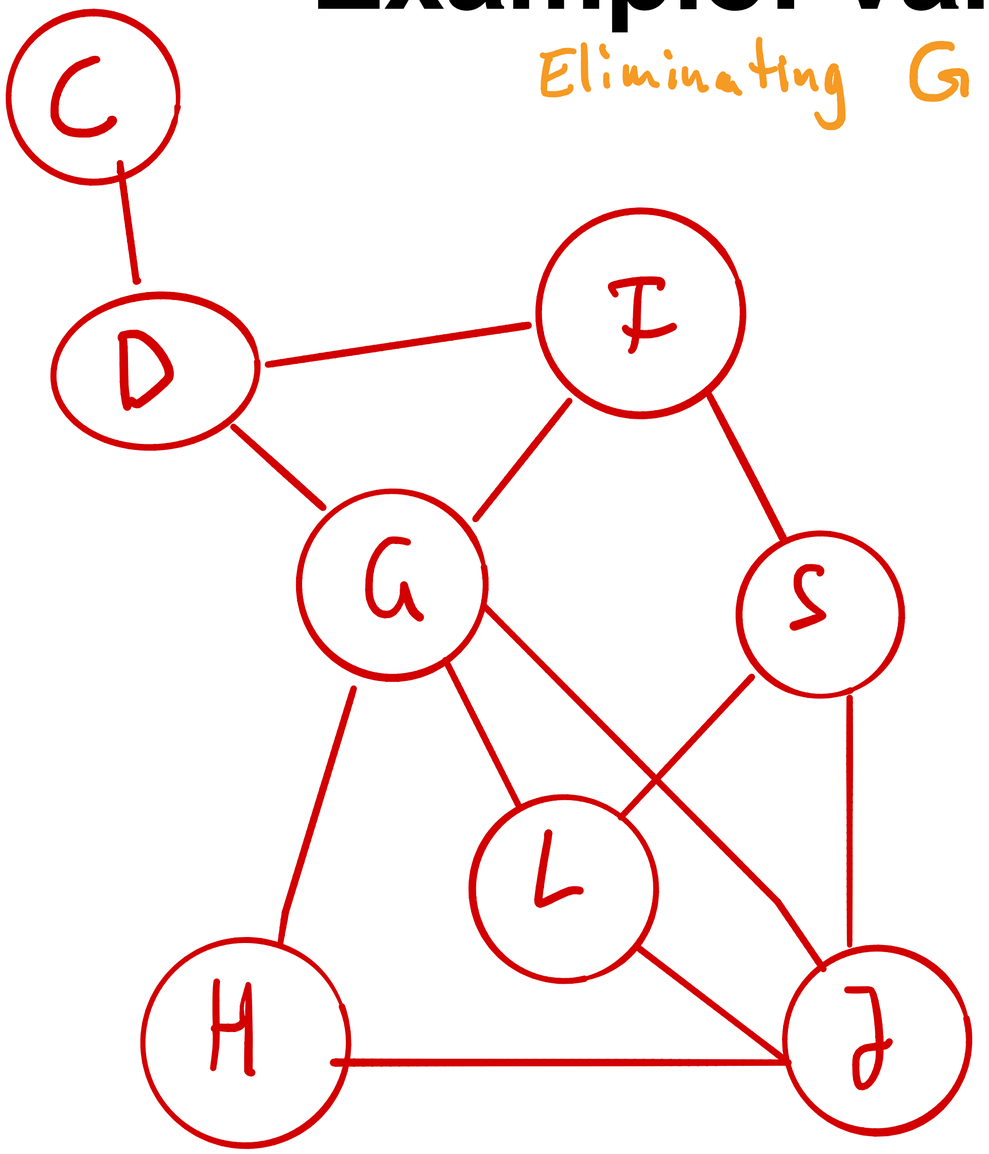
$$= \sum_{L,S} \psi(J,L,S) \cdot \sum_{G} \psi(L,G) \; \phi_3(S,G) \sum_{H} \psi(H,G,J)$$

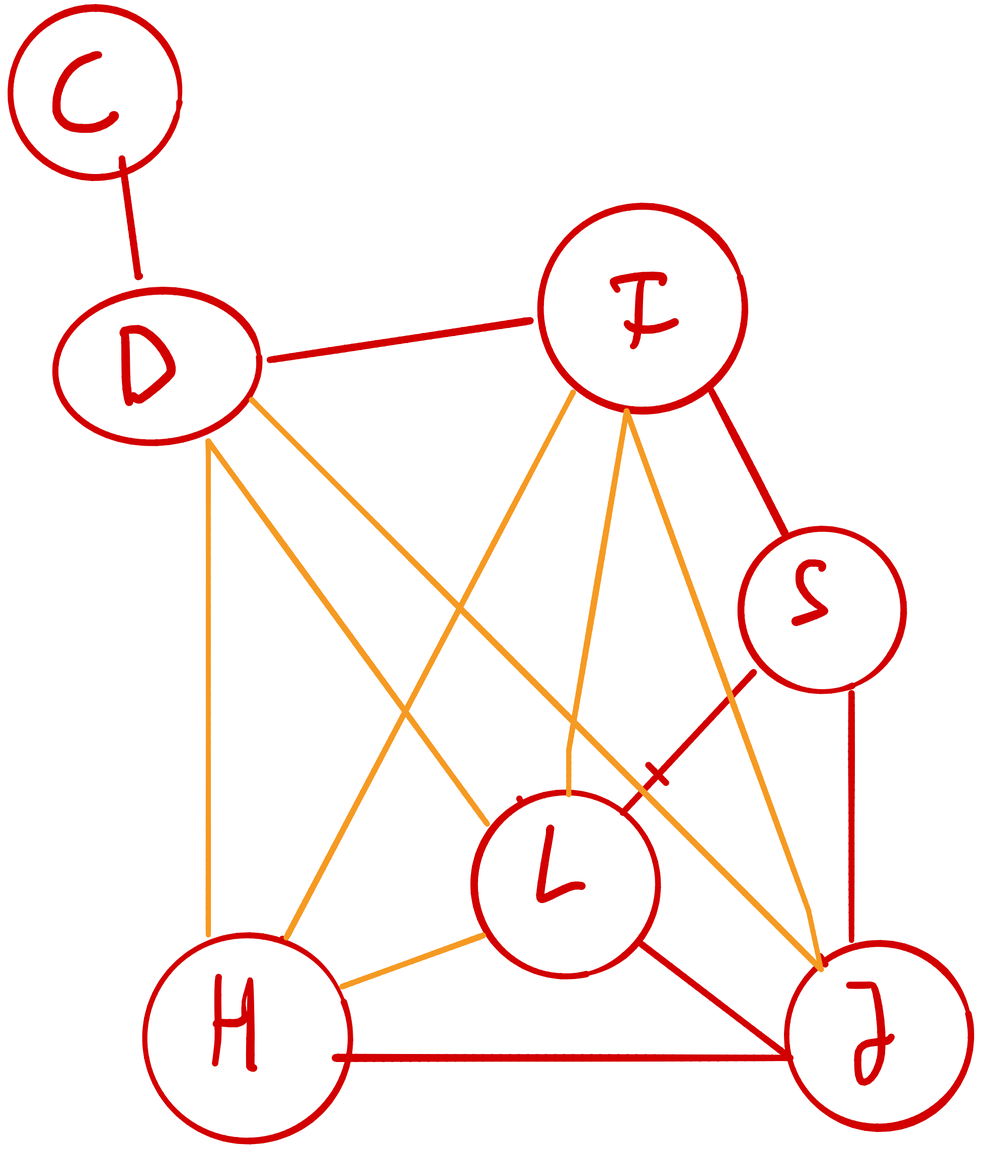
$$= \sum_{L,S} \psi(J,L,S) \cdot \sum_{G} \psi(L,G) \; \phi_3(S,G) \cdot \phi_4(G,J)$$

$$= \sum_{L,S} \psi(J,L,S) \cdot \phi_5(L,S,J)$$

$$= \phi_6(J)$$

Example: Variable Elimination





Variable Elimination Algorithm

- Let (G, p) be BN or MRF. Goal: Compute $p(x_C)$
- Finding a good 'elimination' ordering of the nodes $V = \{v_1, \ldots, v_M\}$:
  - Finding the optimal one is NP-hard (but Variable Elimination has exponential time complexity i.g.).
  - Heuristics for MRFs:
    - Least number of new edges created at each step.
    - Create smallest factor possible at each step.
  - Heuristics for BNs:
    - Iteratively eliminate child-less nodes not in C first. End up with $\mathrm{Anc}^G(C)$, proceed from there.
    - Use heuristics from above.
- Eliminate / Marginalize variables out in that order.
  - by combining all involved factors by summing over all the variables values

Machine Learning 2

Graphical Models
- Exact Inference
- Factor Trees

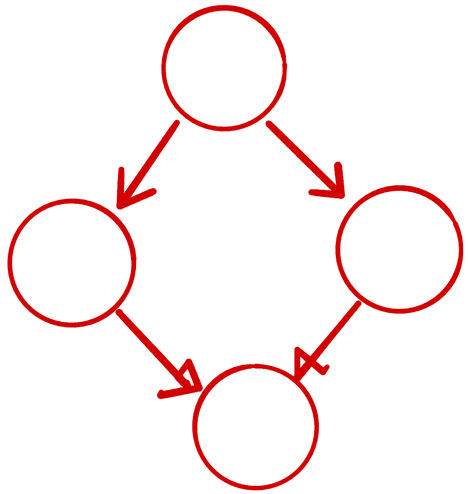
Patrick Forré

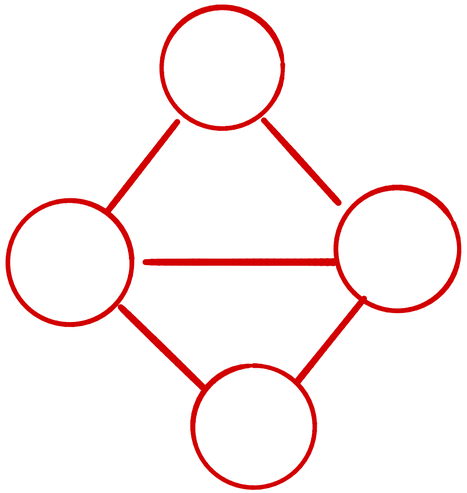
Tree-shaped Factor Graphs

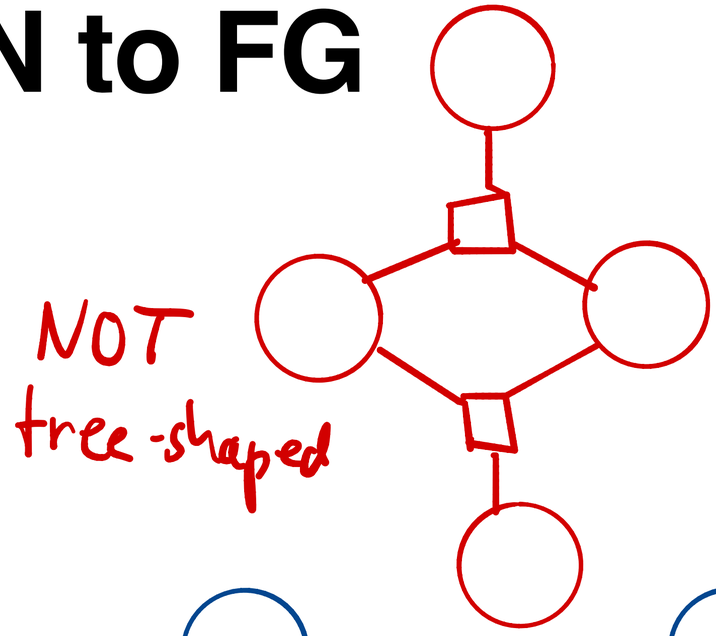
- A Factor Graph $G = (V, F, E)$ is called tree-shaped or Factor Tree if it does not contain any (undirected) cycles.
- Since we can always convert BNs and MRFs to FGs everything said can be applied to BNs and MRFs as long as their FGs are tree-shaped.
- Note that a FG constructed from a BN ('acyclic') might NOT be tree-shaped in general.
- Note that MRFs with cycles might not have cycles anymore in their FGs.

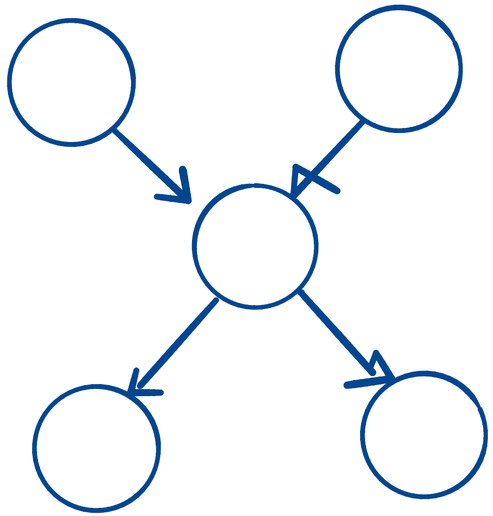
needs to be checked !!

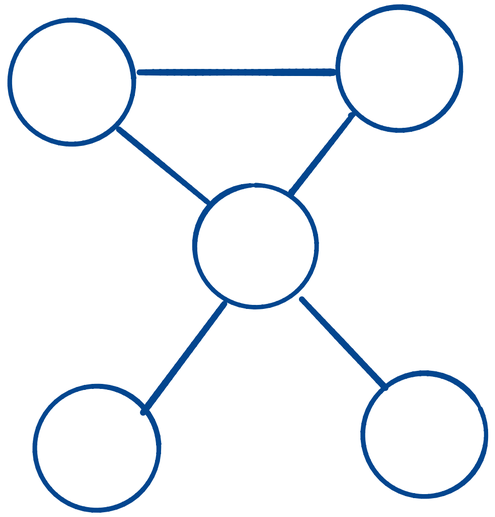
Example: MRF to BN to FG

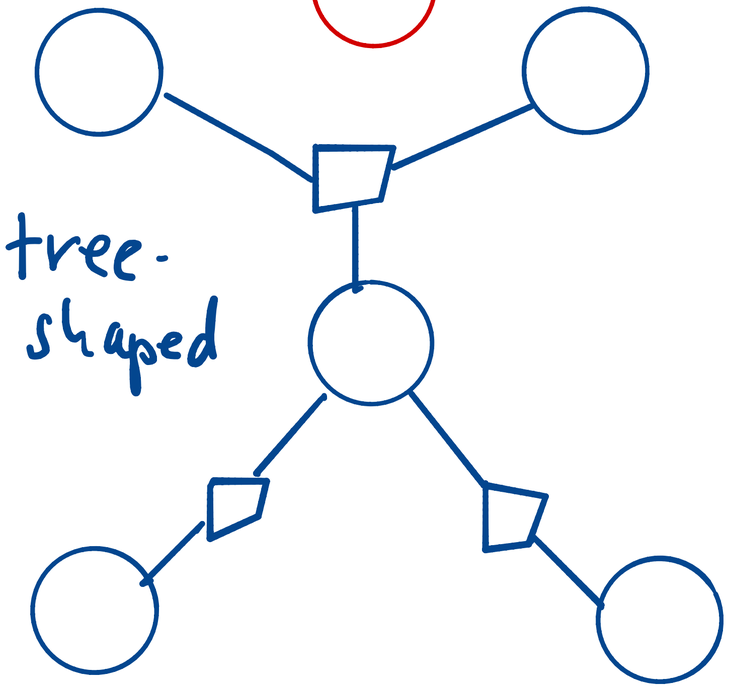












Machine Learning 2

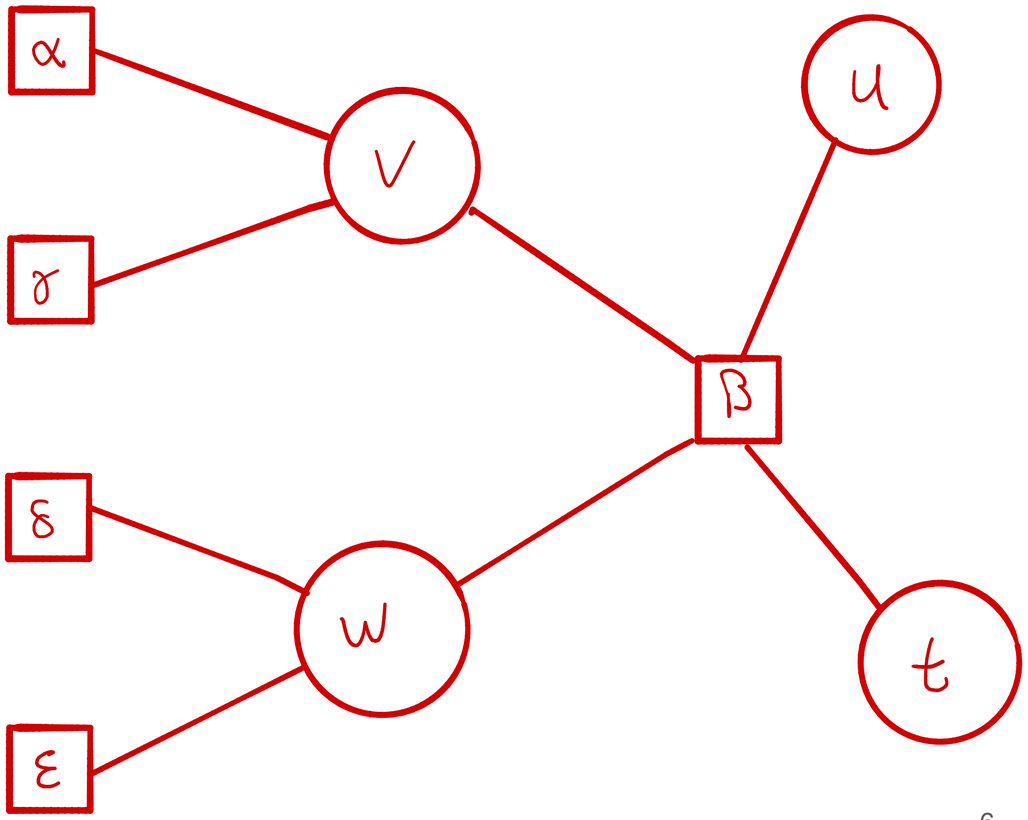
Graphical Models
- Exact Inference
- Sum-Product Algorithm
(in Factor Trees)

Patrick Forré

Example: Factor Tree

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)$.

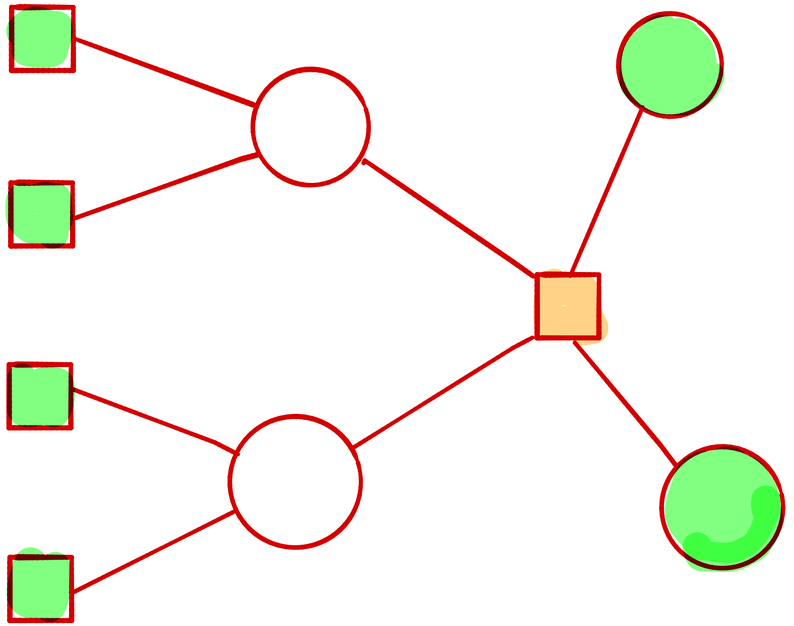
Goal: $p(x_v)$



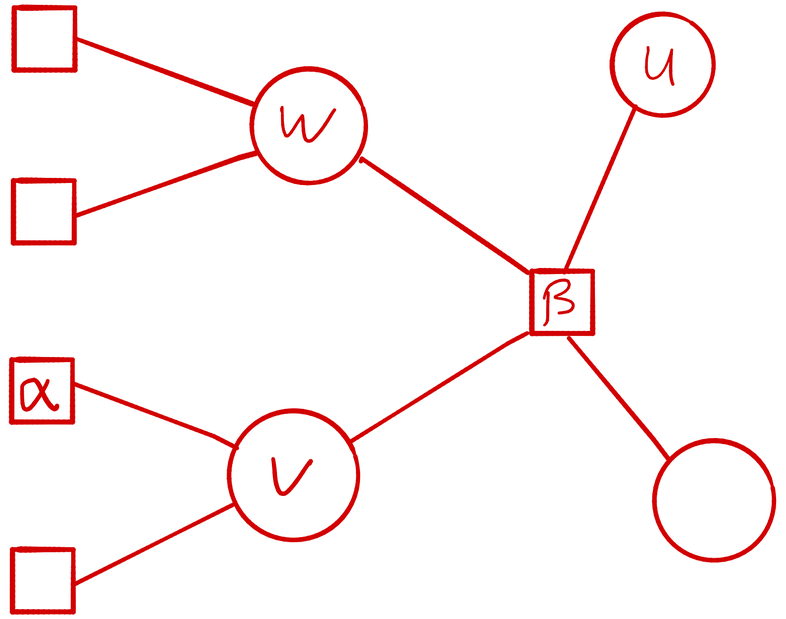
Message Passing

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)$.

- Goal: Find ~~Z and~~ all marginals $p(x_v)$ and $p(x_\alpha)$ for $v \in V$ and $\alpha \in F$.
- Pick a 'Root' node in the middle of G.
- Message Passing (only once forth and back):
  - Pass Messages from Leaves towards Root.
  - Pass Messages from Root back toward Leaves.
- Locally compute 'beliefs' (marginals).

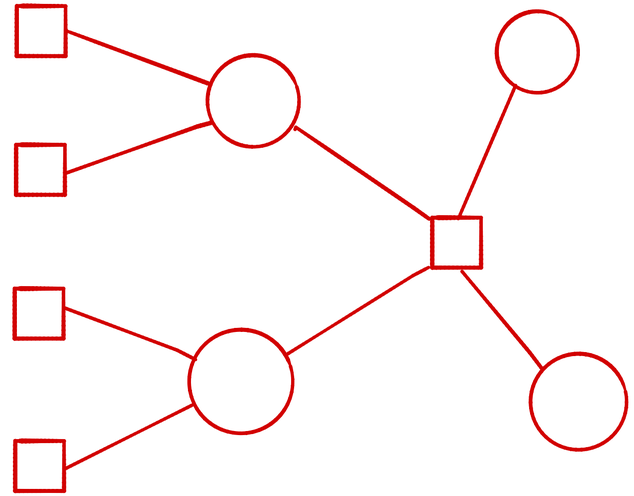


The Messages



- Leaf-variable-to-factor: $\mu_{u\to\beta}(x_u) = 1$
- Non-leaf-variable-to-factor: $\mu_{v\to\beta}(x_v) = \prod_{\gamma\in\partial(v)\setminus\{\beta\}} \mu_{\gamma\to v}(x_v)$
- Leaf-factor-to-variable: $\mu_{\alpha\to v}(x_v) = \psi_\alpha(x_v)$
- Non-leaf-factor-to-variable: $\mu_{\beta\to w}(x_w) = \sum_{x_{\beta\setminus w}} \psi_\beta(x_\beta)\cdot \prod_{u\in\beta\setminus w} \mu_{u\to\beta}(x_u)$

Computing the Beliefs



- Variable beliefs ($v \in V$):
  - $p(x_v) = \frac{1}{Z_v} \cdot \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$
  - $Z_v := \sum_{x_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$
- Factor beliefs ($\alpha \in F$):
  - $p(x_\alpha) = \frac{1}{Z_\alpha} \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$
  - $Z_\alpha := \sum_{x_\alpha} \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$

Exercise: Show that the ‘Beliefs’ are correct

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha).$

- Show that the following equations after the Sum-Product-Algorithm has converged are actually correct:

  - $p(x_v) = \frac{1}{Z_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$ with $Z_v := \sum_{x_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$

  - $p(x_\alpha) = \frac{1}{Z_\alpha} \cdot \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$ with $Z_\alpha := \sum_{x_\alpha} \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$

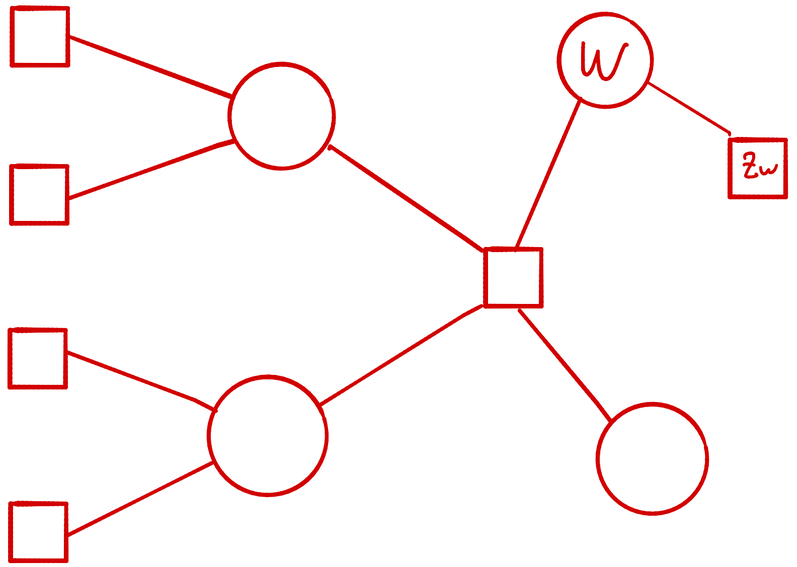
Conditioning via Evidence Factors

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)$.

- Goal: Compute $p(x_v \mid x_C = z_C)$ for some fixed value z_C.

- For this introduce 'evidence factors': $\psi_w(x_w) := \mathbb{1}_{z_w}(x_w)$ for $w \in C$ into a new Factor graph:

$$p(x_{V \setminus C} \mid z_C) \propto \prod_{\alpha \in F} \psi_\alpha(x_\alpha) \cdot \prod_{w \in C} \mathbb{1}_{z_w}(x_w)$$

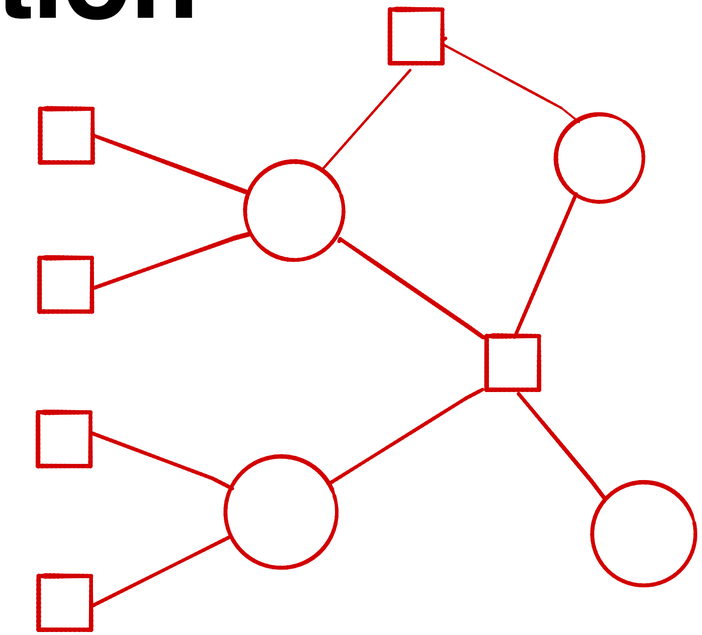


- Run Sum-Product-Algorithm as before.

- Similarly for factor beliefs.

Loopy Belief Propagation

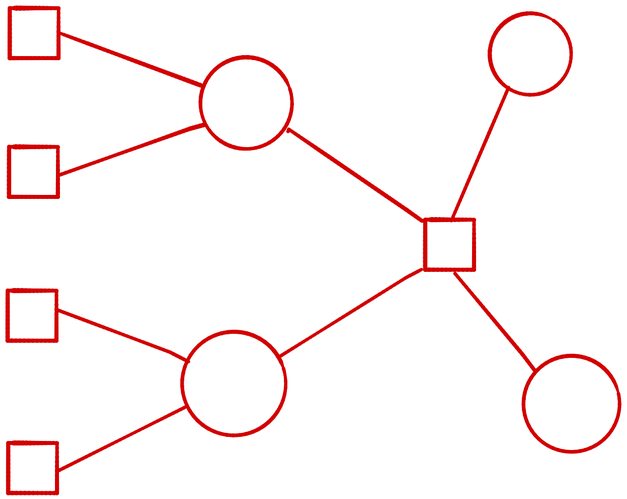
- $G = (V, F, E)$ Factor Graph (possibly non-tree), $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha).$
- Loopy Belief Propagation:
  - Initialize all messages to 1. Then until convergence do:
    - randomly draw node and use usual Sum-Product update rules.
  - Compute bliefs



- Not many theoretical guarantees, not exact anymore (approximate inference)
- In practice reported to converge quickly and approximation often good (but sometimes not).
- Alternatively: Cluster nodes of a cycle togher -> check Junction Tree Algorithm

Sum-Product Algorithm / Belief Propagation

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)$. Pick a 'Root' inside the graph.
- Message Passing (once from Leaves to Root and back):
  - Variable-to-factor: $\mu_{v \to \beta}(x_v) = \prod_{\gamma \in \partial(v) \setminus \{\beta\}} \mu_{\gamma \to v}(x_v)$ Leaf: $\mu_{u \to \beta}(x_u) = 1$
  - Factor-to-variable: $\mu_{\beta \to w}(x_w) = \sum_{x_{\partial(\beta) \setminus \{w\}}} \psi_\beta(x_\beta) \prod_{u \in \partial(\beta) \setminus \{w\}} \mu_{u \to \beta}(x_u)$ Leaf: $\mu_{\alpha \to v}(x_v) = \psi_\alpha(x_v)$
- Compute 'Beliefs' / marginals via:
  - $p(x_v) = \frac{1}{Z_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$ with $Z_v := \sum_{x_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$
  - $p(x_\alpha) = \frac{1}{Z_\alpha} \cdot \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$ with $Z_\alpha := \sum_{x_\alpha} \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$

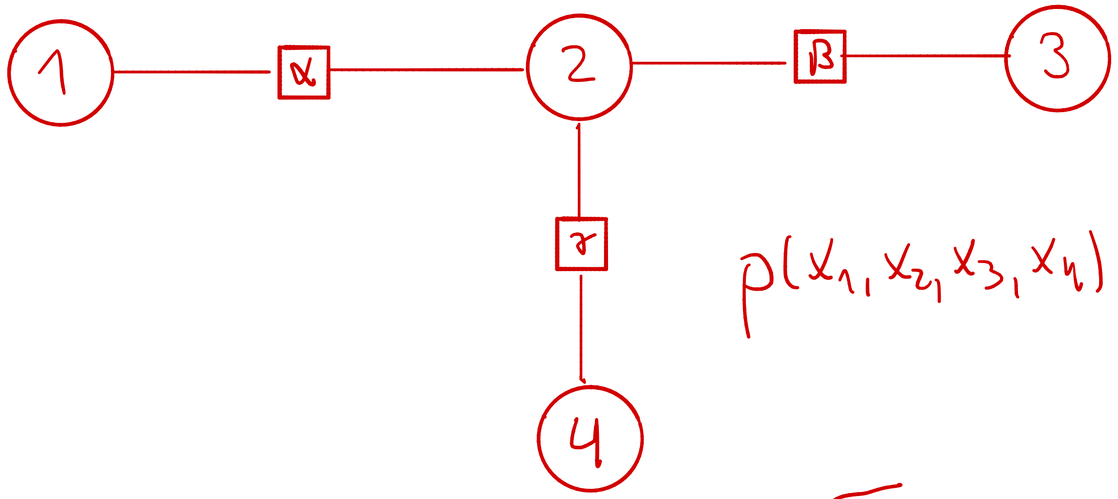


Machine Learning 2

Graphical Models
- Exact Inference
- Sum-Product Algorithm
- Example

Patrick Forré

Example: Sum-Product Algorithm / Belief Propagation



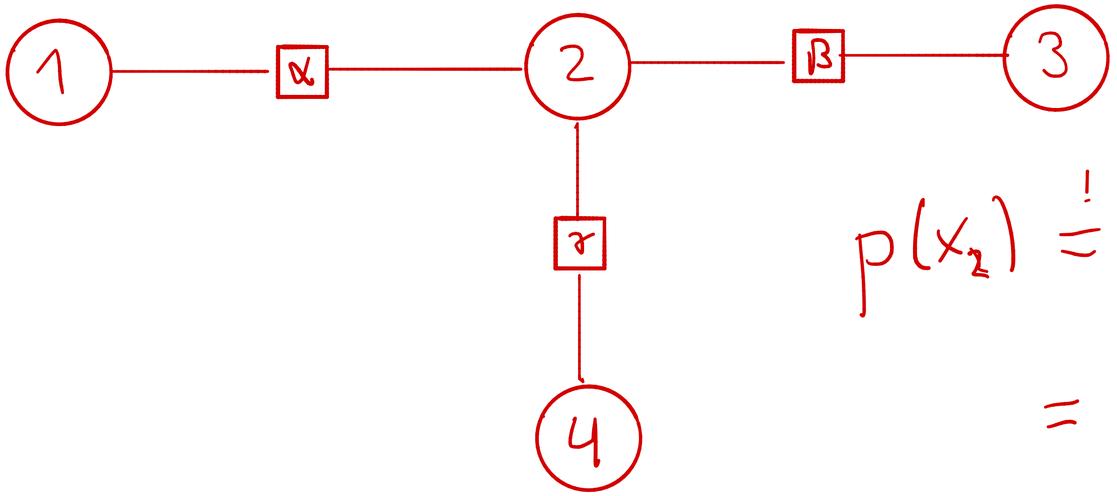
$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} \cdot \psi_\alpha(x_1, x_2)\, \psi_\beta(x_2, x_3)\, \psi_\gamma(x_2, x_4)$$

$$\mu_{1 \to \alpha}(x_1) = 1 \quad , \quad \mu_{\alpha \to 2}(x_2) = \sum_{x_1} \psi_\alpha(x_1, x_2) \cdot \underbrace{\mu_{1 \to \alpha}(x_1)}_{=1} = \sum_{x_1} \psi_\alpha(x_1, x_2)$$

$$\mu_{4 \to \gamma}(x_4) = 1 \quad , \quad \mu_{\gamma \to 2}(x_2) = \sum_{x_4} \psi_\gamma(x_2, x_4) \cdot \underbrace{\mu_{4 \to \gamma}(x_4)}_{=1} = \sum_{x_4} \psi_\gamma(x_2, x_4)$$

$$\mu_{3 \to \beta}(x_3) = 1 \quad , \quad \mu_{\beta \to 2}(x_2) = \sum_{x_3} \psi_\beta(x_2, x_3) \cdot \underbrace{\mu_{3 \to \beta}(x_3)}_{=1} = \sum_{x_3} \psi_\beta(x_2, x_3).$$

Example: Sum-Product Algorithm / Belief Propagation



$$p(x_2) \overset{!}{=} \frac{1}{Z_2} \cdot \mu_{\alpha \to 2}(x_2) \cdot \mu_{\beta \to 2}(x_2) \cdot \mu_{\gamma \to 2}(x_2)$$

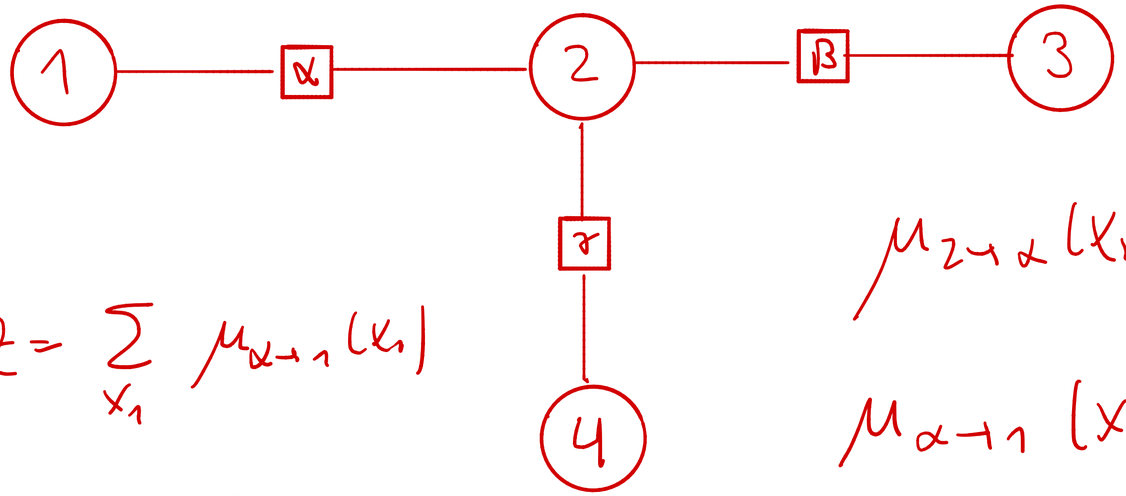
$$= \frac{1}{Z_2} \left(\sum_{x_1} \psi_\alpha(x_1, x_2) \right) \left(\sum_{x_3} \psi_\beta(x_2, x_3) \right) \left(\sum_{x_4} \psi_\gamma(x_2, x_4) \right)$$

$$= \frac{1}{Z_2} \sum_{x_1, x_3, x_4} \psi_\alpha(x_1, x_2) \cdot \psi_\beta(x_2, x_3) \cdot \psi_\gamma(x_2, x_4)$$

$$= \sum_{x_1, x_3, x_4} p(x_1, x_2, x_3, x_4)$$

✓

Example: Sum-Product Algorithm / Belief Propagation



$$\mu_{2\to\alpha}(x_2) = \mu_{\gamma\to 2}(x_2) \cdot \mu_{\beta\to 2}(x_2)$$

$$\mu_{\alpha\to 1}(x_1) = \sum_{x_2} \psi_\alpha(x_1, x_2) \cdot \mu_{2\to\alpha}(x_2)$$

$$Z = \sum_{x_1} \mu_{\alpha\to 1}(x_1)$$

$$p(x_1) \stackrel{?}{=} \frac{1}{Z} \cdot \mu_{\alpha\to 1}(x_1)$$

$$= \frac{1}{Z} \sum_{x_2} \psi_\alpha(x_1, x_2) \cdot \mu_{2\to\alpha}(x_2)$$

$$= \frac{1}{Z} \sum_{x_2} \psi_\alpha(x_1, x_2) \cdot \left(\sum_{x_4} \psi_\gamma(x_2, x_4)\right) \left(\sum_{x_3} \psi_\beta(x_2, x_3)\right)$$

$$= \frac{1}{Z} \sum_{x_2, x_3, x_4} \psi_\alpha(x_1, x_2)\, \psi_\beta(x_2, x_3) \cdot \psi_\gamma(x_2, x_4) = \sum_{x_2, x_3, x_4} p(x_1, x_2, x_3, x_4) \quad \checkmark$$

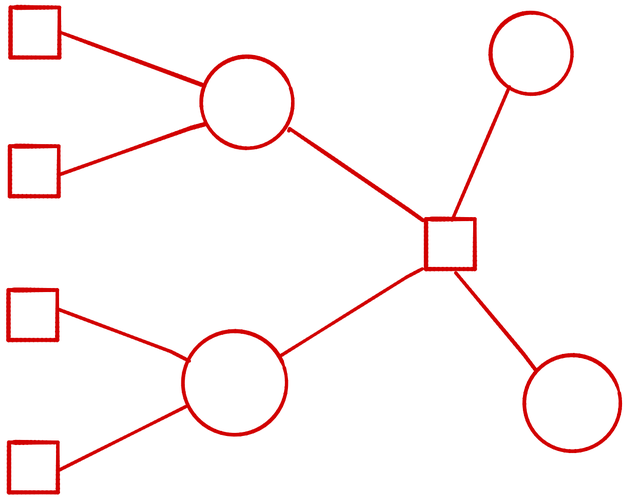
Machine Learning 2

Graphical Models
- Exact Inference
- Sum-Product Algorithm
- Summary

Patrick Forré

Sum-Product Algorithm / Belief Propagation

- $G = (V, F, E)$ Factor Tree with $p(x_V) = \frac{1}{Z} \prod_{\alpha \in F} \psi_\alpha(x_\alpha)$. Pick a 'Root' inside the graph.
- Message Passing (once from Leaves to Root and back):
  - Variable-to-factor: $\mu_{v \to \beta}(x_v) = \prod_{\gamma \in \partial(v) \setminus \{\beta\}} \mu_{\gamma \to v}(x_v)$ Leaf: $\mu_{u \to \beta}(x_u) = 1$
  - Factor-to-variable: $\mu_{\beta \to w}(x_w) = \sum_{x_{\partial(\beta) \setminus \{w\}}} \psi_\beta(x_\beta) \prod_{u \in \partial(\beta) \setminus \{w\}} \mu_{u \to \beta}(x_u)$ Leaf: $\mu_{\alpha \to v}(x_v) = \psi_\alpha(x_v)$
- Compute 'Beliefs' / marginals via:
  - $p(x_v) = \frac{1}{Z_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$ with $Z_v := \sum_{x_v} \prod_{\alpha \in \partial(v)} \mu_{\alpha \to v}(x_v)$
  - $p(x_\alpha) = \frac{1}{Z_\alpha} \cdot \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$ with $Z_\alpha := \sum_{x_\alpha} \psi_\alpha(x_\alpha) \cdot \prod_{v \in \partial(\alpha)} \mu_{v \to \alpha}(x_v)$



Remarks

- For computing conditionals $p(x_v \mid x_C = z_C)$ use Evidence Factors:

$$\psi_w(x_w) := \mathbb{1}_{z_w}(x_w).$$

- For non-tree-shaped factor graphs consider using Loopy Belief Propagation
  - Initialize messages to 1,
  - randomly draw node and update with usual Sum-Product update rules.
- Alternatively, use Junction Tree Algorithm to avoid cycles.