Probabilistic Methods (PM - 330725)

TOPIC 3: Probabilistic Models

Lecture 7

May 2025

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria de Manresa

Roadmap

Learning Outcomes

By the end of this topic, you will be able to:

- update probabilities and make predictions using Bayesian methods and prior distributions.
- model and analyze processes where future states depend on current states, using Markov chains and their properties.
- construct and apply Bayesian networks and Hidden Markov Models for complex system representation.
- use probabilistic models to handle and predict outcomes in sequential data and time series.

Bayesian Networks: graphical recap

- Nodes represent variables
- Arcs/links represent conditional dependence
- We can use the decomposition for any joint distribution.
- The direct / brute-force application generates fully connected graphs
- We can represent much more general relations

Bayesian Networks

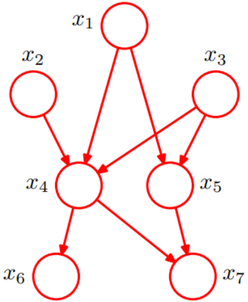
- Nodes represent variables
- Arcs/links represent conditional dependence
- We can use the decomposition for any joint distribution.
- The direct / brute-force application generates fully connected graphs
- We can represent much more general relations

Bayesian Networks

- We can represent relations such as

$$p(x_1)\,p(x_2)\,p(x_3)\,p(x_4 \mid x_1, x_2, x_3)\,p(x_5 \mid x_1, x_3)\,p(x_6 \mid x_4)\,p(x_7 \mid x_4, x_5)$$

- Which is shown below



The general case

- We can think of this as coding the factors

$$p(x_k \mid \mathrm{pa}_k)$$

where pa_k is the set of parents of the variable x_k.

- The inference is then

$$p(\mathbf{x}) = \prod_k p(x_k \mid \mathrm{pa}_k)$$

We will refer to this as factorization.

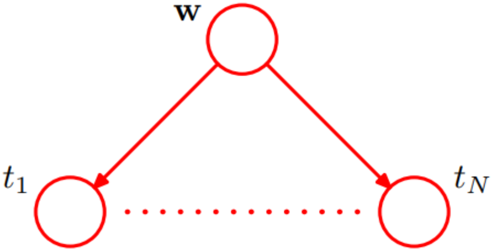
- There can be no directed cycles in the graph.
- The general form is termed a directed acyclic graph — DAG.

Basic example

- We have seen the polynomial regression before

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_n p(t_n \mid \mathbf{w})$$

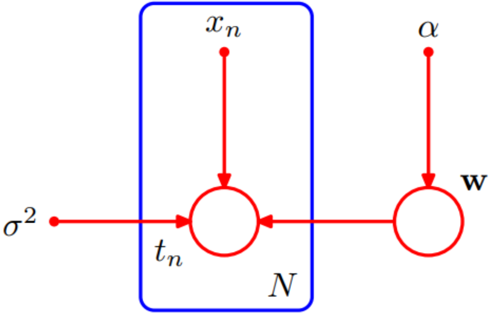
- Which can be visualized as



- We can make the parameters and variables explicit

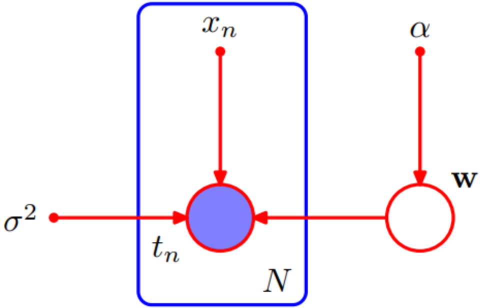
$$p(\mathbf{t}, \mathbf{w} \mid \mathbf{x}, \alpha, \sigma^2) \;=\; p(\mathbf{w} \mid \alpha) \prod_n p(t_n \mid \mathbf{w}, x_n, \sigma^2)$$

as shown here **(Plate Notation)**

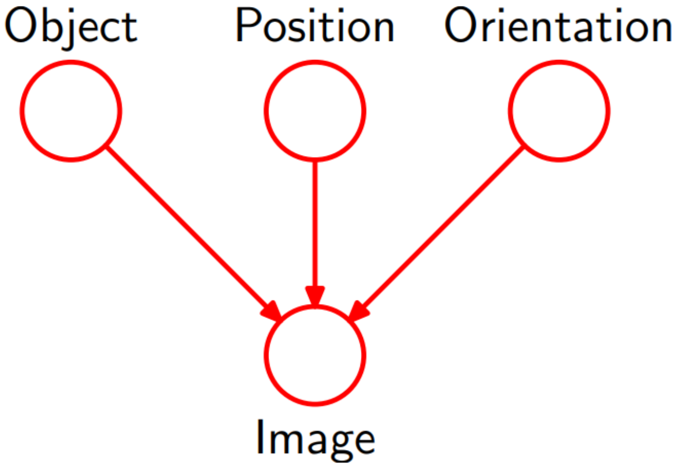


- When entering data we can condition inference on it

$$p(\mathbf{w} \mid \mathbf{t}) \;\propto\; p(\mathbf{w}) \prod_n p(t_n \mid \mathbf{w})$$



Generative Models - Example Image Synthesis



- General joint distribution has $K^2 - 1$ parameters (for K possible outcomes)

$$p(x_1, x_2 \mid \mu) = \prod_{i=1}^{K} \prod_{j=1}^{K} \mu_{ij}^{x_{1i} x_{2j}}$$

- Independent joint distributions have $2(K - 1)$ parameters

$$p(x_1, x_2 \mid \mu) = \prod_{i=1}^{K} \mu_{1i}^{x_{1i}} \prod_{j=1}^{K} \mu_{2j}^{x_{2j}}$$

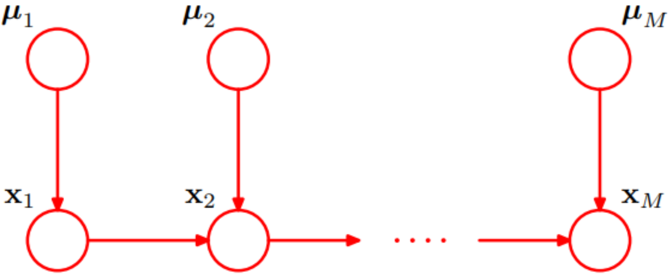
Discrete Variables – 2

- General joint distribution over M variables will have $K^M - 1$ parameters
- A Markov chain with M nodes will have

$$K - 1 \; + \; (M - 1)\, K\, (K - 1)$$

parameters

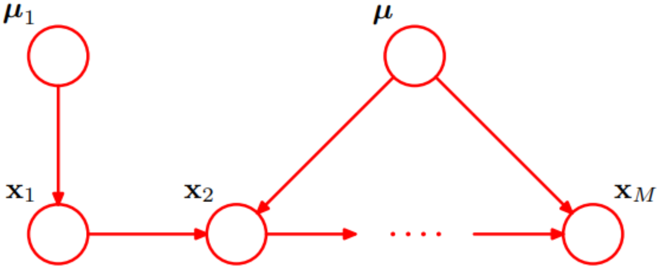
Discrete Variables - Bayesian Parms



- The parameters can be modelled explicitly

$$p(\{x_m, \mu_m\}) = p(x_1 \mid \mu_1)\, p(\mu_1) \prod_{m=2}^{M} p(x_m \mid x_{m-1}, \mu_m)\, p(\mu_m)$$

- It is assumed that $p(\mu_m)$ is a Dirachlet



- For shared parameters the situation is simpler

$$p(\{x_m\}, \mu_1, \mu) = p(x_1 \mid \mu_1)\, p(\mu_1) \prod_{m=2}^{M} p(x_m \mid x_{m-1}, \mu)\, p(\mu)$$

Extension to Linear Gaussian Models

- The model can be extended to have each node as a Gaussian process/variable that is a linear function of its parents

$$p(x_i \mid \mathrm{pa}_i) = \mathcal{N}\left(x_i \,\middle|\, \sum_{j \in \mathrm{pa}_i} w_{ij} x_j + b_i,\; v_i\right)$$

Conditional Independence

- Considerations of independence is important as part of the analysis and setup of a system
- As an example a is independent of b given c

$$p(a \mid b, c) \;=\; p(a \mid c)$$

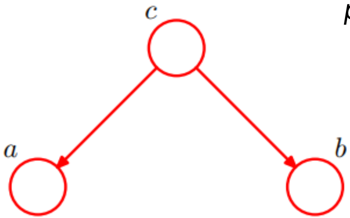
- Or equivalently

$$\begin{aligned} p(a, b \mid c) &= p\big(a \mid b, c\big)\, p(b \mid c) \\ &= p(a \mid c)\, p(b \mid c) \end{aligned}$$

- Frequent notation in statistics

$$a \;\perp\!\!\!\perp\; b \mid c$$

Conditional Independence - Case 1

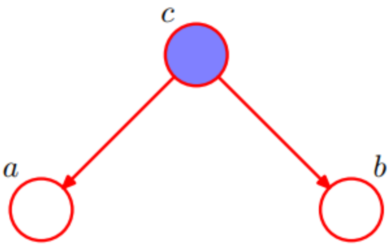


$$p(a, b, c) = p(a \mid c)\, p(b \mid c)\, p(c)$$

$$p(a, b) = \sum_c p(a \mid c)\, p(b \mid c)\, p(c)$$

$$a \perp\!\!\!\perp b \,\big|\, \varnothing$$

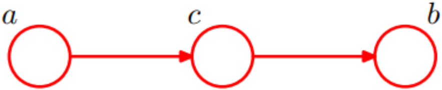
Conditional Independence - Case 1



$$p(a, b \mid c) = \frac{p(a, b, c)}{p(c)}$$

$$= p(a \mid c)\, p(b \mid c)$$

$$a \perp\!\!\!\perp b \mid c$$

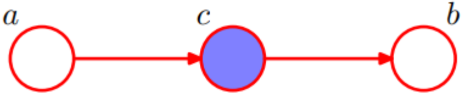


$$p(a,b,c) = p(a)\,p(c \mid a)\,p(b \mid c)$$

$$p(a,b) = p(a)\sum_{c} p(c \mid a)\,p(b \mid c) \;=\; p(a)\,p(b \mid a)$$

$$a \perp\!\!\!\perp b \mid \emptyset$$

Conditional Independence - Case2

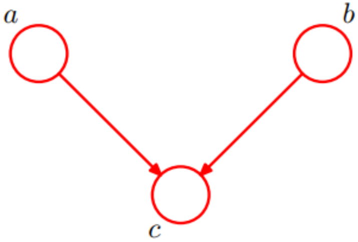


$$p(a, b \mid c) = \frac{p(a, b, c)}{p(c)}$$

$$= \frac{p(a)\, p(c \mid a)\, p(b \mid c)}{p(c)}$$

$$= p(a \mid c)\, p(b \mid c)$$

$$a \perp\!\!\!\perp b \mid c$$



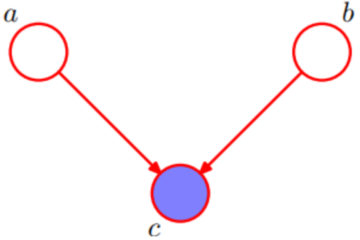
$$p(a, b, c) = p(a)\, p(b)\, p\big(c \mid a, b\big)$$

$$p(a, b) = p(a)\, p(b)$$

$$a \perp\!\!\!\perp b \,\big|\, \varnothing$$

This is the opposite of Case 1 - when c unobserved

Conditional Independence - Case 3



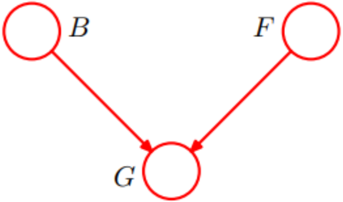
$$p(a, b \mid c) = \frac{p(a, b, c)}{p(c)}$$

$$= \frac{p(a)\, p(b)\, p(c \mid a, b)}{p(c)}$$

$$a \perp\!\!\!\perp b \,\big|\, c$$

This is the opposite of Case 1 - when c unobserved

Diagnostics - Out of fuel?



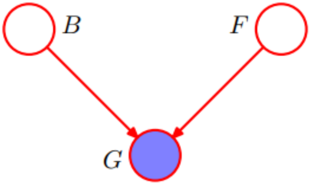
$$p(G = 1 \mid B = 1,\ F = 1) \ = \ 0.8$$
$$p(G = 1 \mid B = 1,\ F = 0) \ = \ 0.2$$
$$p(G = 1 \mid B = 0,\ F = 1) \ = \ 0.2$$
$$p(G = 1 \mid B = 0,\ F = 0) \ = \ 0.1$$

B = Battery $\qquad p(B = 1) = 0.9$

F = Fuel Tank $\qquad p(F = 1) = 0.9$

G = Fuel Gauge $\Longrightarrow \quad p(F = 0) = 0.1$

Diagnostics - Out of fuel?



$$p(F = 0 \mid G = 0) = \frac{p(G = 0 \mid F = 0)\, p(F = 0)}{p(G = 0)} \approx 0.257$$

Observing $G = 0$ increased the probability of an empty tank.

Diagnostics - Out of fuel? Compute $p(F = 0 \mid G = 0)$

Using Bayes' Rule:

$$p(F = 0 \mid G = 0) = \frac{p(G = 0 \mid F = 0) \cdot p(F = 0)}{p(G = 0)}$$

First compute the numerator terms:
Marginalize over B:

$$\begin{aligned} p(G = 0 \mid F = 0) &= p(G = 0 \mid B = 1, F = 0) \cdot p(B = 1) + p(G = 0 \mid B = 0, F = 0) \cdot p(B = 0) \\ &= 0.8 \cdot 0.9 + 0.9 \cdot 0.1 = 0.72 + 0.09 = 0.81 \end{aligned}$$

So numerator:

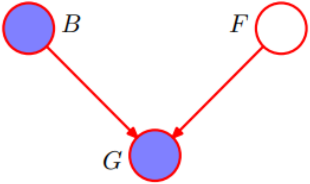
$$p(G = 0 \mid F = 0) \cdot p(F = 0) = 0.81 \cdot 0.1 = 0.081$$

Now compute full marginal $p(G = 0)$:

$$\begin{aligned} p(G = 0) &= \sum_{B,F} p(G = 0 \mid B, F) \cdot p(B) \cdot p(F) \\ &= 0.2 \cdot 0.9 \cdot 0.9 + 0.8 \cdot 0.9 \cdot 0.1 + 0.8 \cdot 0.1 \cdot 0.9 + 0.9 \cdot 0.1 \cdot 0.1 \\ &= 0.162 + 0.072 + 0.072 + 0.009 = 0.315 \end{aligned}$$

$$p(F = 0 \mid G = 0) = \frac{0.081}{0.315} \approx 0.257$$

Answer: $\boxed{p(F = 0 \mid G = 0) \approx 0.257}$



$$p(F = 0 \mid G = 0,\ B = 0) \ = \ \frac{p(G = 0 \mid B = 0,\ F = 0)\, p(F = 0)}{\sum_F p(G = 0 \mid B = 0,\ F)\, p(F)} \ \approx \ 0.111$$

Observing $B = 0$ implies a less likely empty tank.

Diagnostics - Out of fuel? Compute $p(F = 0 \mid G = 0, B = 0)$

Use Bayes again:

$$p(F = 0 \mid G = 0, B = 0) = \frac{p(G = 0 \mid B = 0, F = 0) \cdot p(F = 0)}{\sum_F p(G = 0 \mid B = 0, F) \cdot p(F)}$$

Numerator:

$$p(G = 0 \mid B = 0, F = 0) = 0.9, \quad p(F = 0) = 0.1 \Rightarrow 0.9 \cdot 0.1 = 0.09$$

Denominator:

$$0.9 \cdot 0.1 + 0.8 \cdot 0.9 = 0.09 + 0.72 = 0.81$$

So:

$$p(F = 0 \mid G = 0, B = 0) = \frac{0.09}{0.81} \approx 0.111$$

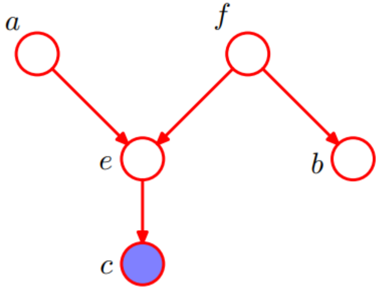
Answer: $\boxed{p(F = 0 \mid G = 0, B = 0) \approx 0.111}$

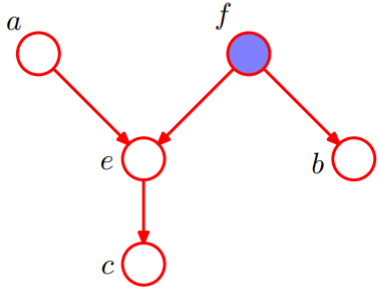
D-separation

- Consider non-intersecting subsets A, B, and C in a directed graph
- A path between subsets A and B is considered blocked if it contains a node such that:
    1. the arcs are head-to-tail or tail-to-tail and the node is in the set C
    2. the arcs meet head-to-head and neither the node nor its descendants are in the set C
- If all paths between A and B are blocked, then A is d-separated from B by C.
- If there is d-separation, then all the variables in the graph satisfy

$$A \perp\!\!\!\perp B \,\big|\, C$$

D-separation





Inference in graphical models

- Consider inference of $p(x, y)$; we can formulate this as

$$p(x, y) = p(x \mid y)\, p(y) = p(y \mid x)\, p(x)$$

- We can further marginalise

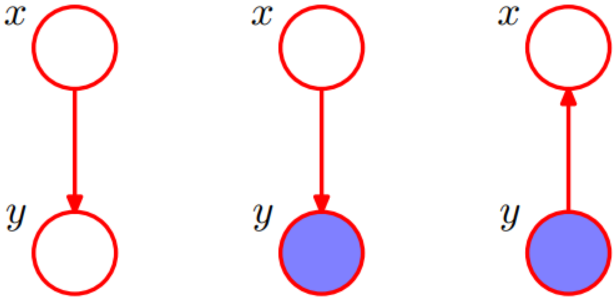
$$p(y) = \sum_{x'} p\big(y \mid x'\big)\, p\big(x'\big)$$

- Using Bayes' rule we can reverse the inference

$$p(x \mid y) = \frac{p(y \mid x)\, p(x)}{p(y)}$$

- Helpful as mechanisms for inference

Inference in graphical models



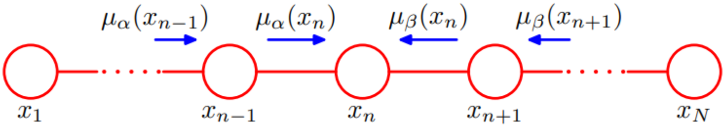
(a) (b) (c)

Inference on a chain

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N)$$

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \sum_{x_N} p(\mathbf{x})$$

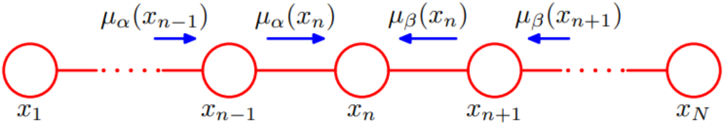
Inference on a chain



$$p(x_n) = \frac{1}{Z}\left[\underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right]\right]}_{\mu_a(x_n)}\right.$$

$$\left.\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right]\right]}_{\mu_b(x_n)}\right]$$

Inference on a chain

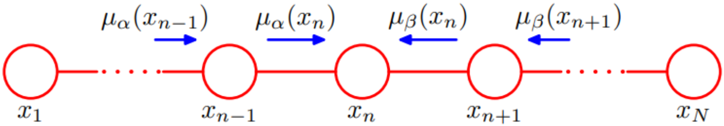


$$\mu_a(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_1} \psi_{1,2}(x_1, x_2) \right]$$

$$= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n)\, \mu_a(x_{n-1})$$

$$\mu_b(x_n) = \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right]$$

$$= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1})\, \mu_b(x_{n+1})$$



$$\mu_a(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \qquad \mu_b(x_{N-1}) = \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)$$

$$Z = \sum_{x_n} \mu_a(x_n)\, \mu_b(x_n)$$

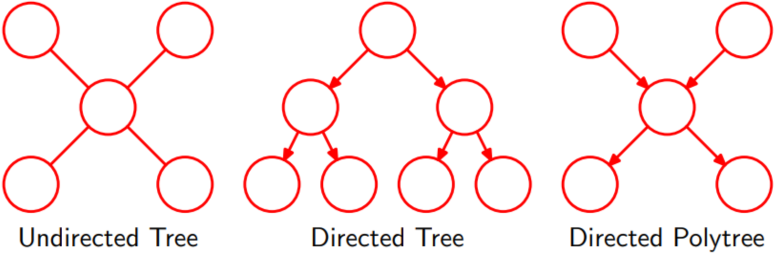
Inference on a chain

- To compute local marginals
  - Compute and store forward messages $\mu_a(x_n)$
  - Compute and store backward messages $\mu_b(x_n)$
  - Compute Z at all nodes
  - Compute

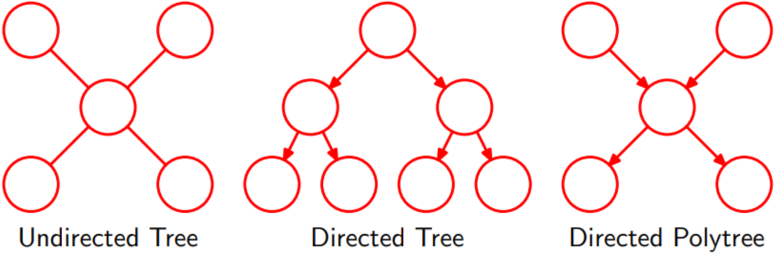
$$p(x_n) = \frac{1}{Z} \mu_a(x_n) \, \mu_b(x_n)$$

    for all variables

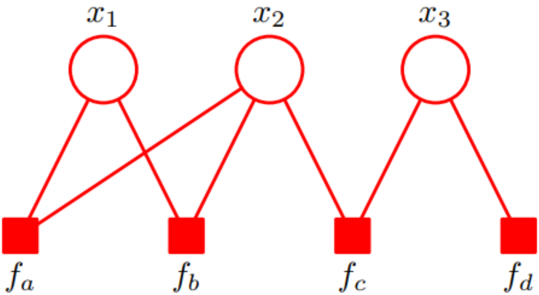
Inference in trees



Factor Graphs



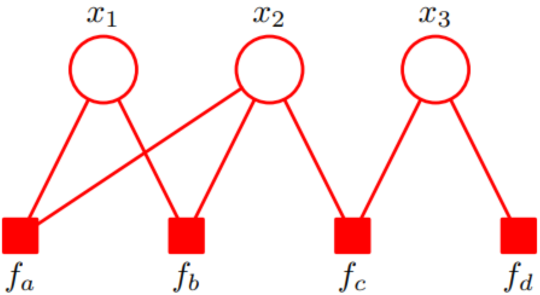
Factor Graphs



$$p(\mathbf{x}) = f_a(x_1, x_2)\, f_b(x_1, x_2)\, f_c(x_2, x_3)\, f_d(x_3)$$

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

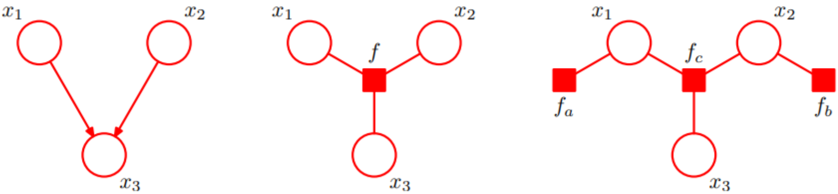
Factor Graphs from Directed Graphs



$$p(\mathbf{x}) = f_a(x_1, x_2)\, f_b(x_1, x_2)\, f_c(x_2, x_3)\, f_d(x_3)$$

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s)$$

Factor Graphs from Directed Graphs



$$p(\mathbf{x}) = p(x_1)\, p(x_2)$$

$$p(x_3 \mid x_1, x_2)$$

$$f(x_1, x_2, x_3) =$$

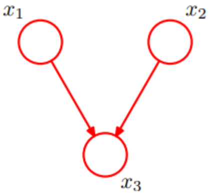
$$p(x_1)\, p(x_2)\, p(x_3 \mid x_1, x_2)$$

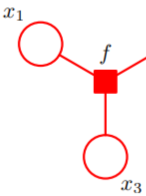
$$f_a(x_1) = p(x_1)$$

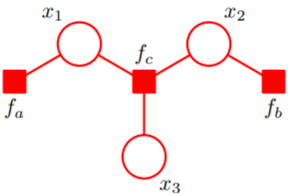
$$f_b(x_2) = p(x_2)$$

$$f_c(x_1, x_2, x_3) = p(x_3 \mid x_2, x_1)$$

Factor Graphs from Undirected Graphs







$$\psi(x_1, x_2, x_3)$$

$$f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$$

$$f_a(x_1, x_2, x_3) f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$$

The Sum-Product Algorithm

Objective

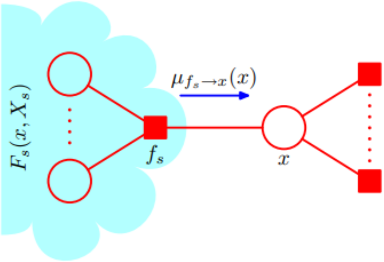
- exact, efficient algorithm for computing marginals
- allow multiple marginals to be computed efficiently

Key Idea

- The distributive law

$$ab + ac \;=\; a\left(b + c\right)$$

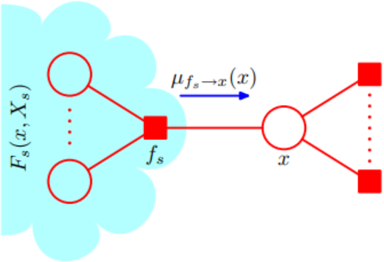
The Sum-Product Algorithm



$$p(x) = \sum_{x \setminus x} p(x)$$

$$p(x) = \prod_{s \in Ne(x)} F_s(x, X_s)$$

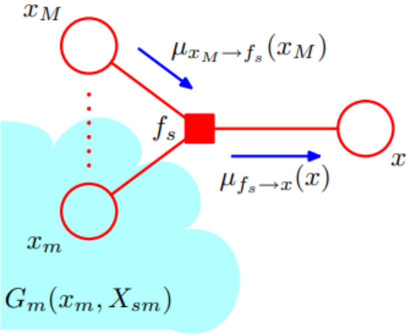
The Sum-Product Algorithm



$$p(x) = \prod_{s \in Ne(x)} \left[\sum_{X_s} F_s(x, X_s) \right] = \prod_{s \in Ne(x)} \mu_{f_s \to x}(x)$$

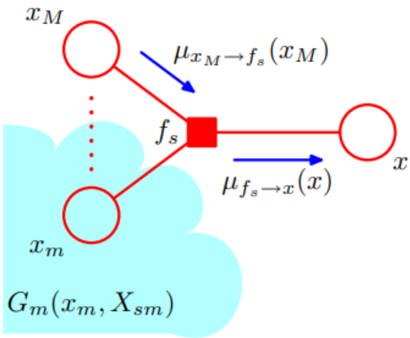
$$\mu_{f_s \to x}(x) = \sum_{X_s} F_s(x, X_s)$$

The Sum-Product Algorithm



$$F_s(x, X_s) = f_s(x, x_1, \ldots, x_M)\, G_1(x_1, X_{s1}) \cdots G_M(x_M, X_{sM})$$

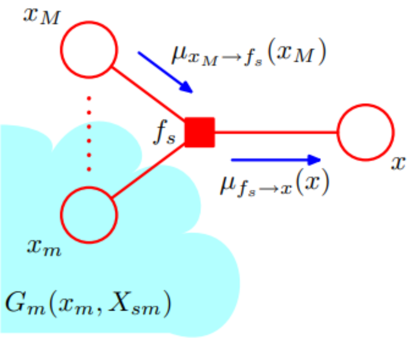
The Sum-Product Algorithm



$$\mu_{f_s \to x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in Ne(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right]$$

$$= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in Ne(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

The Sum-Product Algorithm



$$\mu_{x_m \to f_s}(x_m) \sum_{X_{sm}} G_m(x_m, X_{sm}) = \sum_{X_{sm}} \prod_{l \in Ne(x_m) \setminus f_s} F_l(x_m, X_{lm}) =$$

$$\prod_{l \in Ne(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

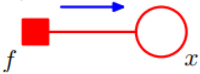
Initialization

- For variable nodes

$$\mu_{x \to f}(x) = 1$$

- For factor nodes

$$\mu_{f \to x}(x) = f(x)$$



The Sum-Product Algorithm

- To compute local marginals
  - Pick an arbitrary node as root
  - Compute and propagate messages from leafs to root (store messages)
  - Compute and propagate messages from root to leaf nodes (store messages)
  - Compute products of received messages and normalise as required
- Propagate up the tree and down again to compute all marginals (as needed)

Wrapping up

Today we discussed:

1. Bayesian Networks
2. Conditional Independence
3. Inference in graphical models
4. Factor Graphs
5. TheSum-Product Algorithm

Some References

- Pratt J., Raiffa H., Schlaifer R., Introduction to Statistical Decision Theory, The MIT Press, 2008
- Hoffmann M. D., Gelman A., The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, arXiv:1111.4246, 2011
- A. Gelman, J. B. Carlin, H. S. Stern, Bayesian Data Analysis, CRC Press, 2013
- Walsh B., Markov Chain Monte Carlo and Gibbs Sampling, Lecture Notes for EEB 596z, 2002
- R. A. Howard, Dynamic Programming and Markov Process, The MIT Press, 1960
- Rabiner L. R., A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77.2, 1989
- W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrik, 57/1, 04/1970
- Kevin B. Korb, Ann E. Nicholson, Bayesian Artificial Intelligence, CRC Press, 2010 Pearl J., Causality, Cambridge University Press, 2009
- L. E. Baum, T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, The Annals of Mathematical Statistics, 37, 1966

Thank you very much!

ANY QUESTIONS OR COMMENTS?