Probabilistic Graphical Models (I)

Hongxin Zhang
zhx@cad.zju.edu.cn

State Key Lab of CAD&CG, ZJU
2015-03-31

Probabilistic Graphical Models

- Modeling many real-world problems => a large number of random variables
  - Dependences among variables may be used to reduce the size to encode the model (PCA ?), or
  - They may be the goal by themselves, that is, the idea is to understand the correlations among variables.

Modeling the domain

- Discrete random variables
  - Take 5 random binary variables (A, B, C, D, E)
  - i.i.d. data from a multinomial distribution

A	B	C	D	E
a	$\sim b$	$\sim c$	$\sim d$	$\sim e$
a	b	$\sim c$	d	$\sim e$
a	$\sim b$	c	d	$\sim e$

Goals

- (Parameter) Learning: using training data, estimate the joint distribution
  - Which are the values $P(A, B, C, D, E)$,?
  - ... and if there were one hundred binary variables? (Size of model certainly greater than number of atoms on Earth!)

- Inference: Given the distribution $P(A, B, C, D, E)$,
  - Belief updating: compute the probability of an event
    - What is the probability of $A=a$ given $E=e$?
  - Maximum a posterior: compute the states of variables that maximize their probability.
    - Which state of A maximizes $P(A \mid E=e)$? Is it a or $\sim a$?

The unstructured approach

- To specify the joint distribution, there is an exponential number of values:

$$p(a, b, c, d, e), p(a, b, c, d, \neg e), p(a, b, c, \neg d, e),$$

$$p(a, b, c, \neg d, \neg e), p(a, b, \neg c, d, e), p(a, b, \neg c, d, \neg e),$$

$$p(a, b, \neg c, \neg d, e), p(a, b, \neg c, \neg d, \neg e), \ldots$$

- We can compute the probability of events by:

$$p(a) = \sum_{B,C,D,E} p(a, B, C, D, E)$$

$$p(a|d, \neg e) = \frac{p(a, d, \neg e)}{p(d, \neg e)} = \frac{\sum_{B,C} p(a, B, C, d, \neg e)}{\sum_{A,B,C} p(A, B, C, d, \neg e)}$$

- There are exponentially many terms in the summations...

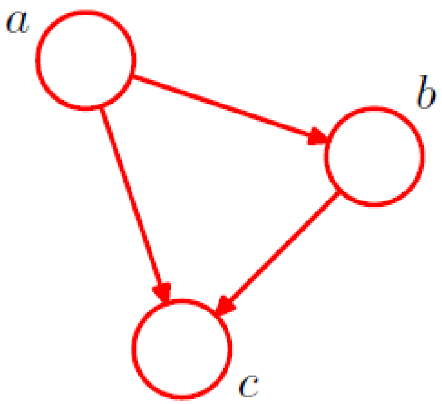
The naïve Bayesian approach

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

- Application: Email spanning

Bayesian Networks

- An arbitrary **joint distribution** $p(a, b, c)$ over three variables a, b, and c
  - the product rule of probability:



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

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

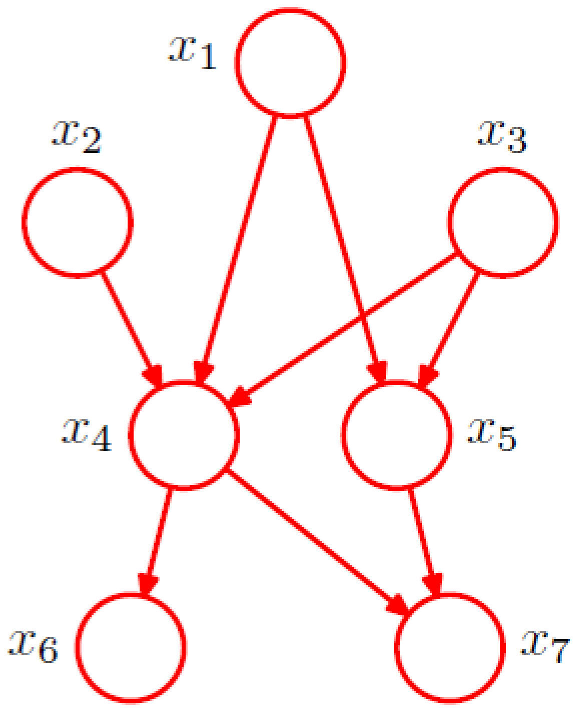
- General case: $p(x_1, x_2, \ldots, x_K)$

$$p(x_1, \ldots, x_K) = p(x_K | x_1, \ldots, x_{K-1}) \ldots p(x_2 | x_1) p(x_1)$$

Not fully connected graph

- Joint distribution: $p(x_1,\ x_2, \ldots, x_7)$

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



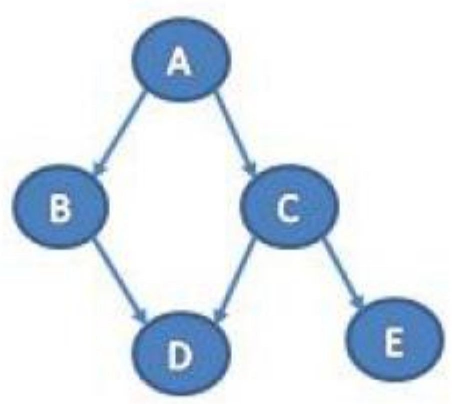
General form

- For a graph with K nodes, *the joint distribution* is given by:

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

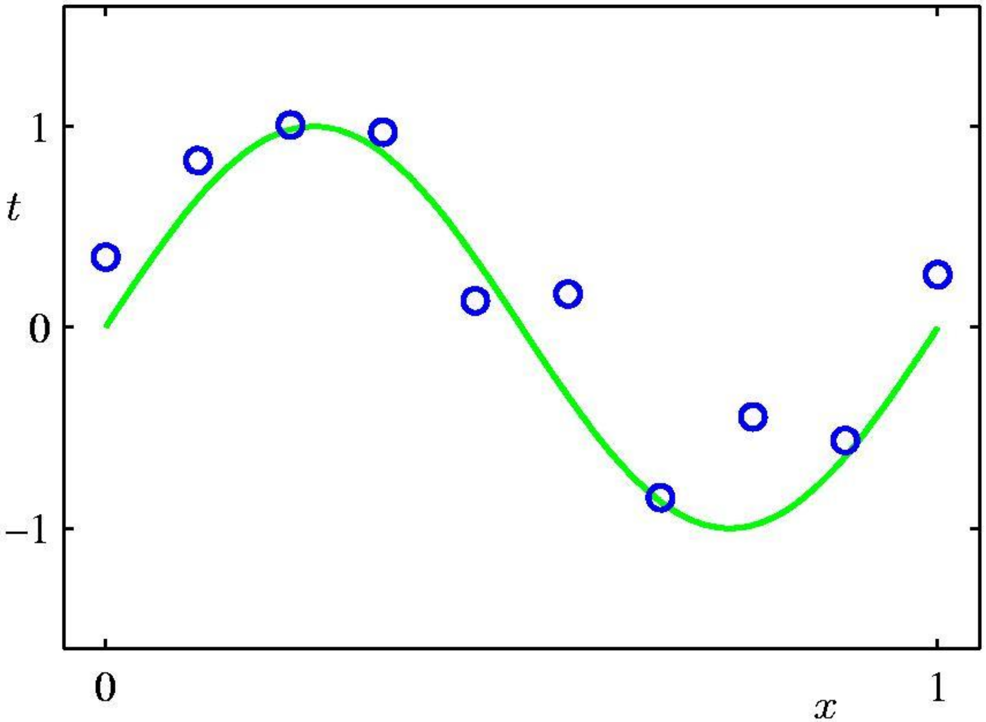
- where pa_k denotes the set of parents of x_k, *and* $\boldsymbol{x} = \{x_1, \ldots, x_K\}$

Definitions



- A set of variables associated with nodes of a Directed Acyclic Graph (DAG).
  - Markov condition (w.r.t. the DAG): each variable is independent of its non-descendants given its parents.
  - For each variable (node), local probability distributions:
    - *P(A), P(B|A=a), P(B|A=a), P(C|A=a), P(C|A=~a), P(D|b, c), P(D|~b, c), P(D|~b,c); P(D|~b,~c), P(E|c), P(E|~c),*
  - All these values are precise.

Regression revisit: Polynomial Curve Fitting



$$\mathbf{h}(x) = \begin{pmatrix} h_0(x) \\ h_1(x) \\ \vdots \\ h_M(x) \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{pmatrix} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix}$$

$$t = \mathbf{h}(x) \cdot \mathbf{w}$$

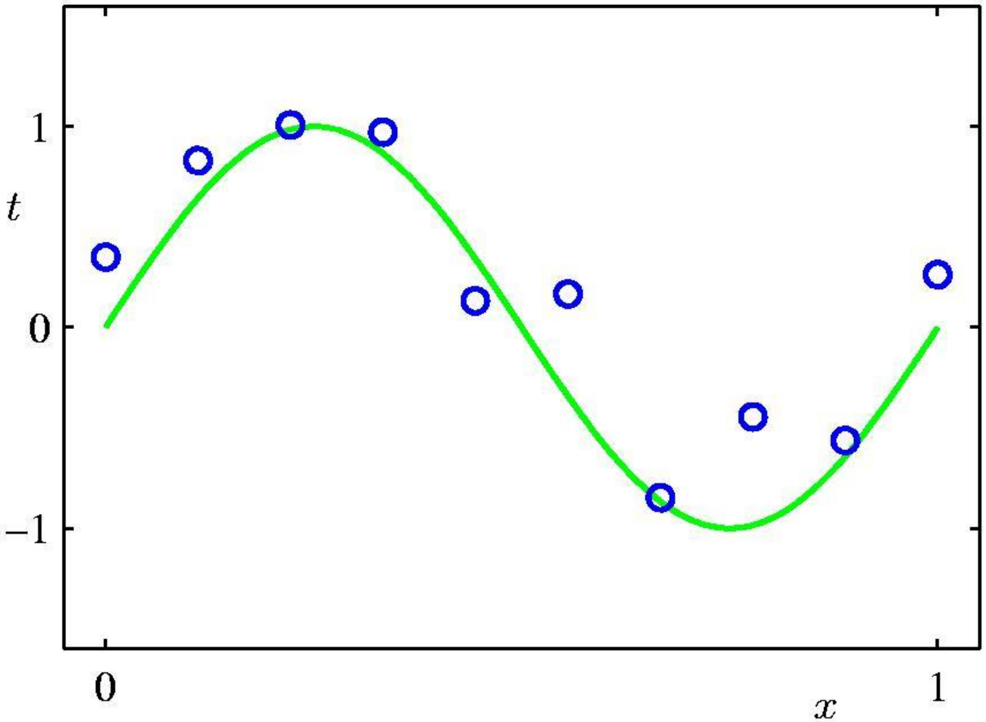
$$\mathbf{w} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{t}$$

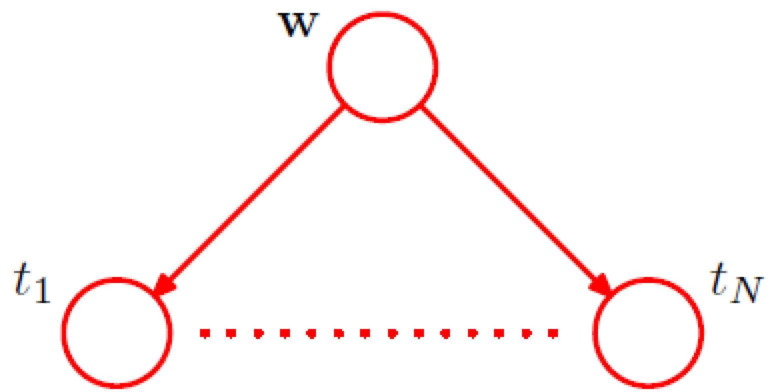
Normal equation

$$t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x)$$

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n|\mathbf{w})$$

Example: Polynomial regression

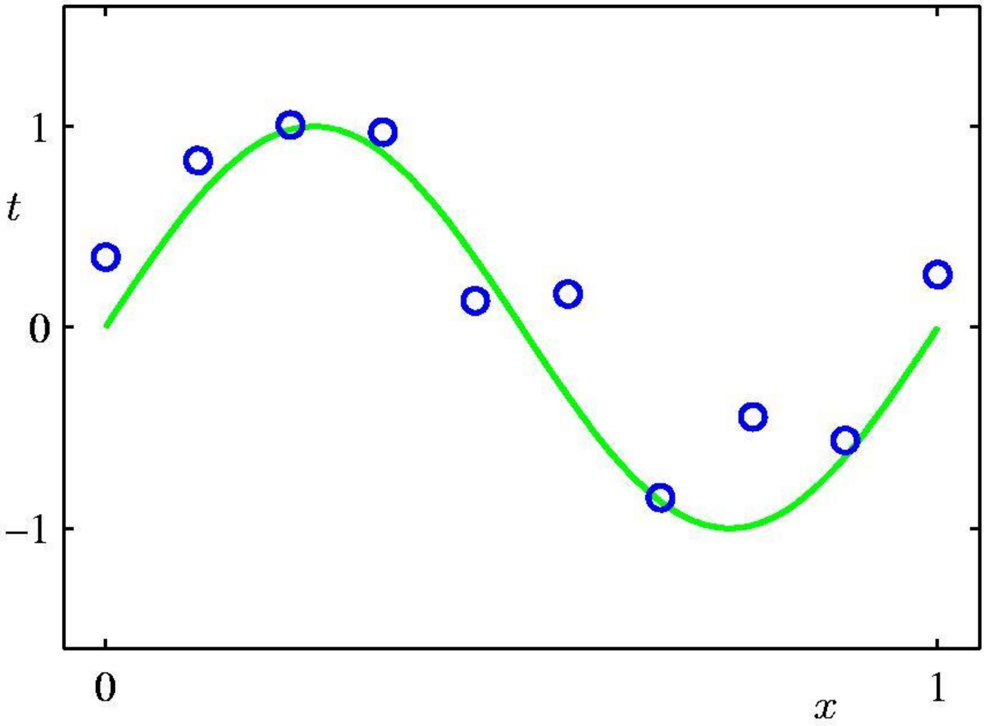


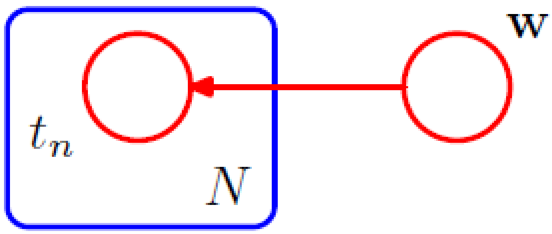


$$t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x)$$

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n | \mathbf{w})$$

Example: Polynomial regression

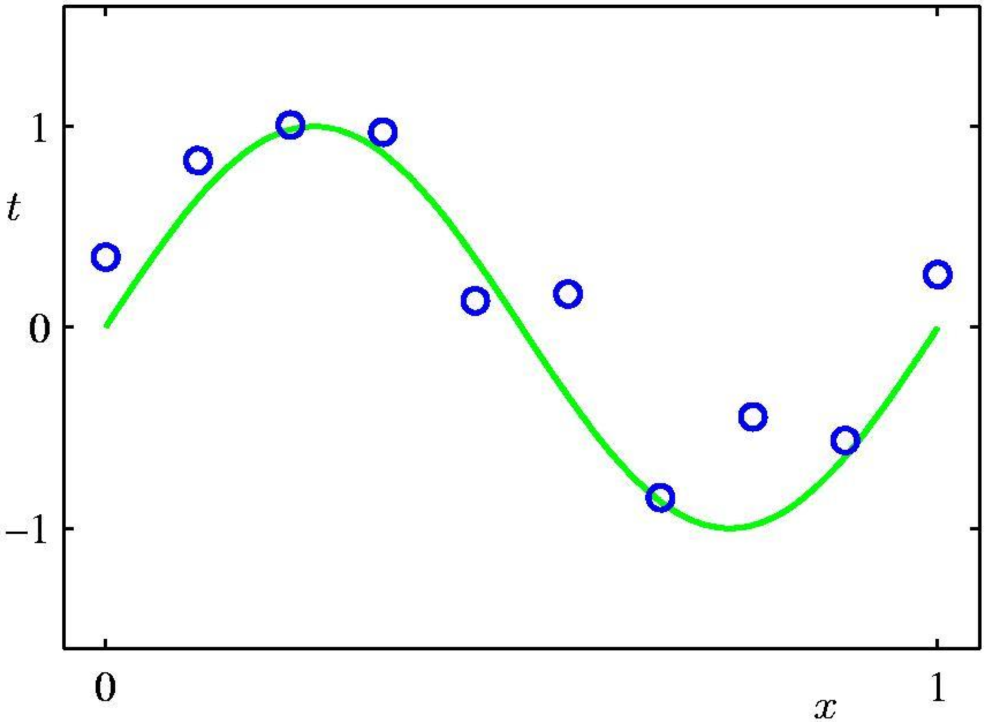


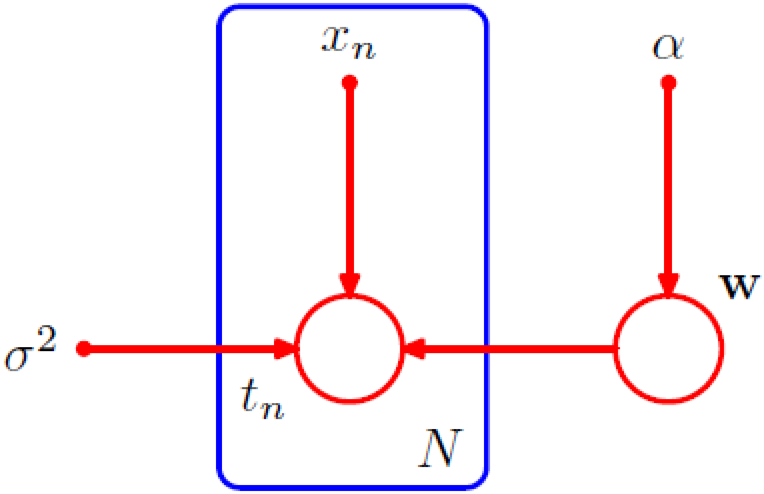


$$t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x)$$

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n | \mathbf{w})$$

Example: Polynomial regression





$$t(x, \mathbf{w}) = w_0 + w_1 h_1(x) + w_2 h_2(x) + \ldots + w_N h_N(x) = \sum_{j=0}^{N} w_j h_j(x)$$

$$p(\mathbf{t}, \mathbf{w} | \mathbf{x}, \alpha, \sigma^2) = p(\mathbf{w}|\alpha) \prod_{n=1}^{N} p(t_n | \mathbf{w}, x_n, \sigma^2)$$

the noise variance *σ2, and the hyperparameter α* representing the precision of the Gaussian prior over **w**

Linear-Gaussian models

- Consider an arbitrary DAG over *D* variables in which node i represents a single continuous random variable x_i having a *Gaussian distribution*

- The mean of this distribution is taken to be a linear combination of the states of its parent nodes pa_i of node i

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

Linear-Gaussian models

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

$$\begin{aligned}
\ln p(\mathbf{x}) &= \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}_i) \\
&= -\sum_{i=1}^{D} \frac{1}{2v_i}\left(x_i - \sum_{j \in \mathrm{pa}_i} w_{ij} x_j - b_i\right)^2 + \mathrm{const}
\end{aligned}$$

Linear-Gaussian models

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

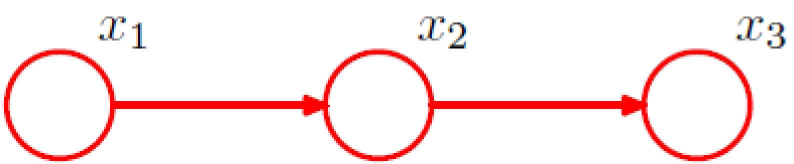
$$x_i = \sum_{j\in\mathrm{pa}_i} w_{ij}x_j + b_i + \sqrt{v_i}\epsilon_i \qquad \mathbb{E}[x_i] = \sum_{j\in\mathrm{pa}_i} w_{ij}\mathbb{E}[x_j] + b_i$$

$$\begin{aligned}
\mathrm{cov}[x_i, x_j] &= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])\right] \\
&= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left\{\sum_{k\in\mathrm{pa}_j} w_{jk}(x_k - \mathbb{E}[x_k]) + \sqrt{v_j}\epsilon_j\right\}\right] \\
&= \sum_{k\in\mathrm{pa}_j} w_{jk}\mathrm{cov}[x_i, x_k] + I_{ij}v_j
\end{aligned}$$

Linear-Gaussian models

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

- Case 1: no links in the graph
  - The joint distribution:
    - 2*D* parameters and represents
    - *D* independent univariate Gaussian distributions.
- Case 2: fully connected graph
  - *D*(*D*-1)/2+*D* independent parameters
- Case 3:



$$\boldsymbol{\mu} = (b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1)^{\mathrm{T}}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} v_1 & w_{21}v_1 & w_{32}w_{21}v_1 \\ w_{21}v_1 & v_2 + w_{21}^2 v_1 & w_{32}(v_2 + w_{21}^2 v_1) \\ w_{32}w_{21}v_1 & w_{32}(v_2 + w_{21}^2 v_1) & v_3 + w_{32}^2(v_2 + w_{21}^2 v_1) \end{pmatrix}$$

Conditional independence

- Three random variables: a, b and c
  - a is conditionally independent of b given c iff
  - $P(a \mid b, c) = P(a \mid c)$

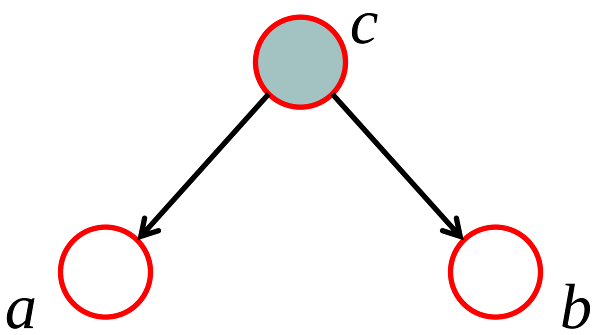
$$\boxed{a \perp\!\!\!\perp b \mid c}$$

- This can be re-written in following way
  - $P(a, b \mid c) = P(a \mid b, c)\, P(b \mid c)$

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

The joint distribution of a and b factorizes into the product of the marginal distribution of a and ~b.

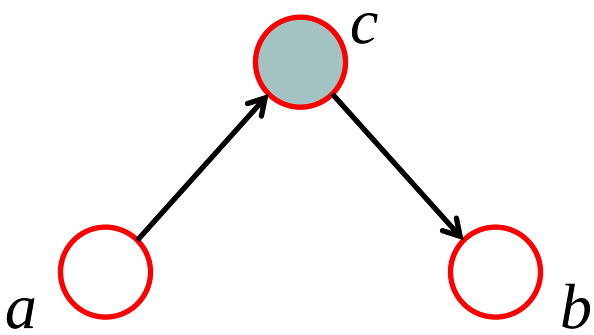
Simple example (1)



- Joint distribution:
  - $P(a, b, c) = P(a \mid c)\, P(b \mid c)\, P(c)$

- Condition on c:
  - $P(a, b \mid c) = P(a, b, c) / P(c) = P(a \mid c)\, P(b \mid c)$
  - => $a \perp\!\!\!\perp b \mid c$

Simple example (2)



- Joint distribution:
  - $P(a, b, c) = P(a)\,P(c \mid a)\,P(b \mid c)$ $\boxed{a \perp\!\!\!\perp b \mid c}$
- Factorization:

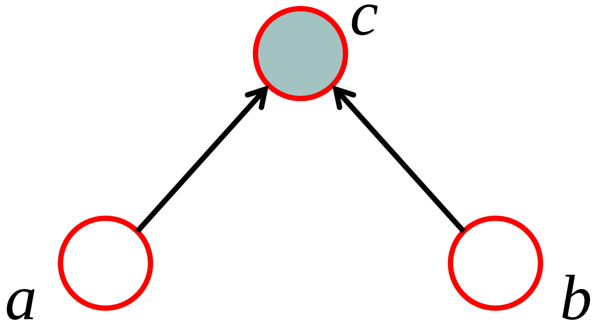
$$P(a,b) = \sum_c P(a,b,c) = P(a)\sum_c P(c \mid a)P(b \mid c)$$
$$= P(a)P(b \mid a)$$

- Condition on c:

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

Bayesian Theorem

Simple example (3)



- Joint distribution:
  - $P(a, b, c) = P(a)\, P(b)\, P(c \mid a, b)$ $\quad \cancel{\boxed{a \perp\!\!\!\perp b \mid c}}$
- Factorization:

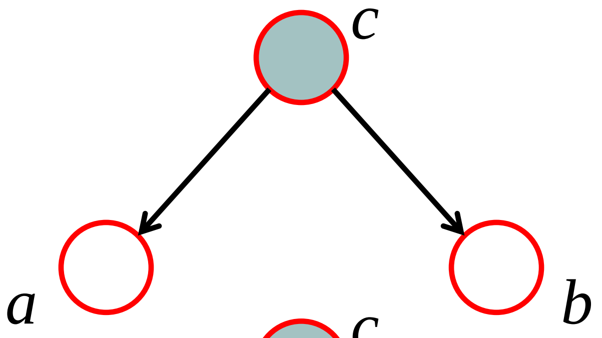
$$P(a,b) = \sum_c P(a,b,c) = P(a)P(b)\sum_c P(c \mid a,b)$$
$$= P(a)P(b)$$

- Condition on c:

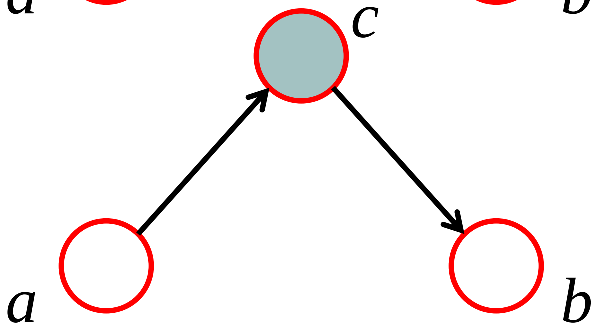
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)P(b)P(c \mid a,b)}{P(c)}$$
$$\neq P(a \mid c)P(b \mid c)$$

Conditional independence

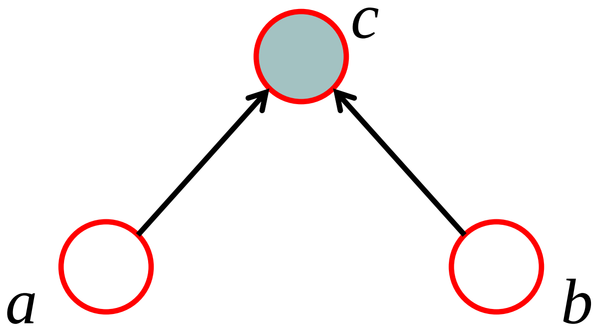
- Tail-to-Tail: yes



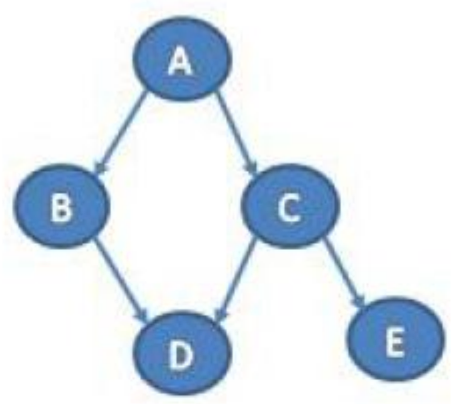
- Head-to-Tail: yes



- Head-to-Head: no



Markov condition



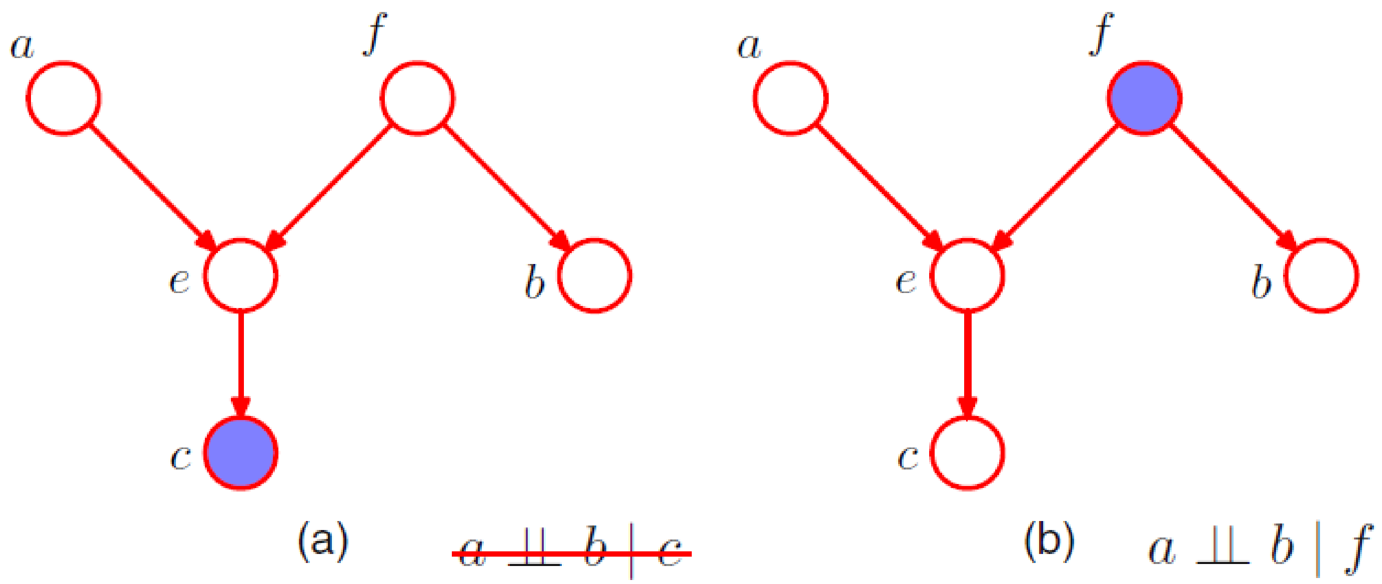
- We say that node *y* is a *descendant* of node *x* if there is a path from x to y in which each step of the path follows the directions of the arrows.
- If each variable is independent of its non-descendants given its parents, then:

$$B \perp\!\!\!\perp (C, E) | A,$$
$$D \perp\!\!\!\perp (A, E) | (B, C),$$
$$E \perp\!\!\!\perp (A, B, D) | C.$$

D-separation

- All possible paths from any node in *A* to any node in *B.* Any such path is said to be *blocked* if it includes a node such that either
  - the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set *C,* or
  - the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set *C*
- If all paths are blocked, then *A* is said to be *d-separated* from *B* by *C.*

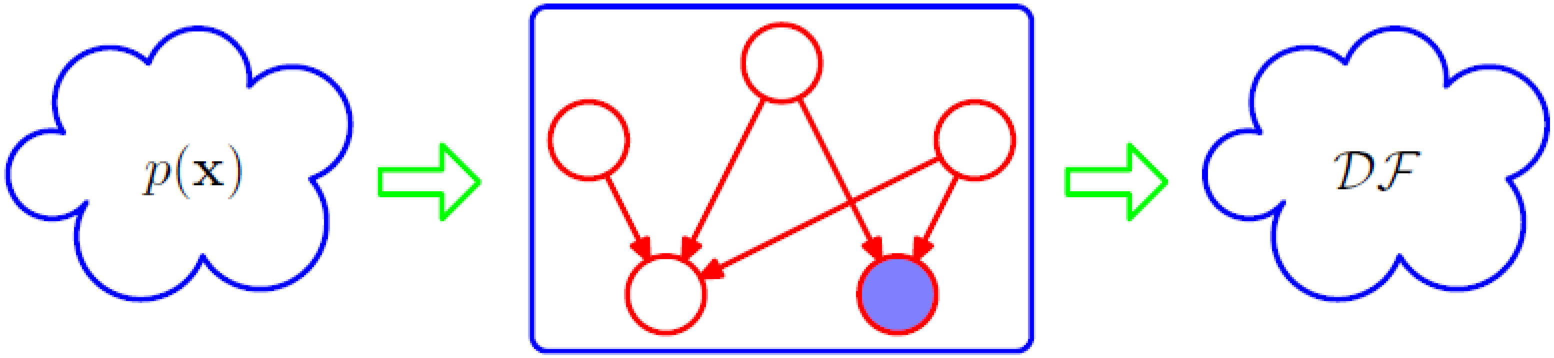
D-separation



- In graph (a), the path from a *to* b is *not blocked* by node c
- In graph (b), the path from a *to* b is *blocked* by node f and e

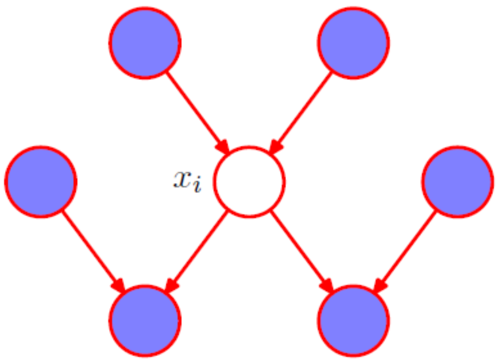
D-separation

- A particular directed graph represents a specific decomposition of a joint probability distribution into a product of conditional probabilities
- A directed graph is a filter



Markov blanket

- Joint distribution $p(x_1, \ldots, x_D)$ represented by a directed graph having *D* nodes
  - $$p(\mathbf{x}_i|\mathbf{x}_{\{j\neq i\}}) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_D)}{\int p(\mathbf{x}_1, \ldots, \mathbf{x}_D)\,\mathrm{d}\mathbf{x}_i} = \frac{\prod_k p(\mathbf{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\mathbf{x}_k|\mathrm{pa}_k)\,\mathrm{d}\mathbf{x}_i}$$



- The set of nodes comprising the parents, the children and the co-parents is called the Markov blanket

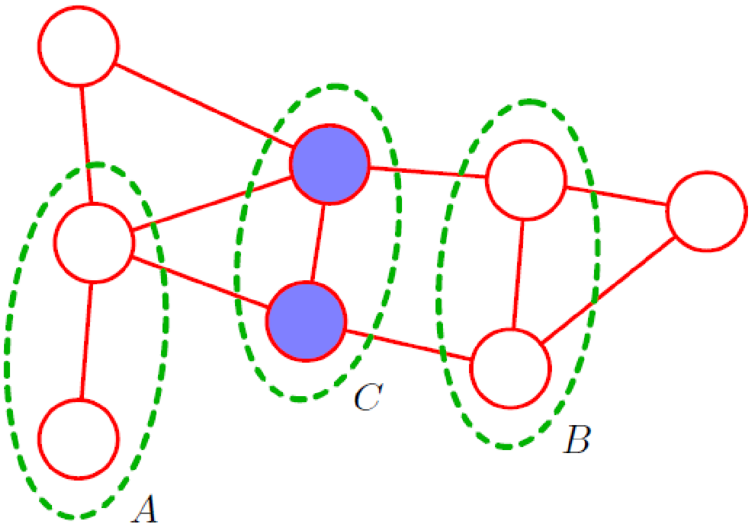
Markov Random Fields

- Also known as a *Markov network or an undirected graphical model*

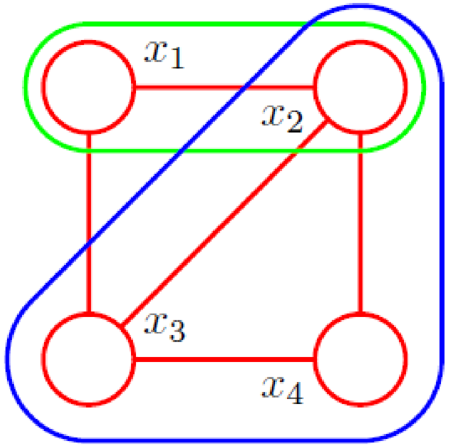
- Conditional independence properties:

Conditional dependence exists if there exists a path that connects any node in *A to any* node in *B.*

If there are no such paths, then the conditional independence property must hold.



Clique



- A subset of the nodes in a graph such that there exists a link between all pairs of nodes in the subset
  - In other words, the set of nodes in a clique is fully connected
  - Maximal clique …
  - A four-node undirected graph showing a clique (outlined in green) and a maximal clique (outlined in blue)

Potential function

- $\boldsymbol{x}_C$: the set of variables in that clique C
- The joint distribution is written as a product of *potential functions* $\psi_C(\boldsymbol{x}_C)$ over the maximal cliques of the graph

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

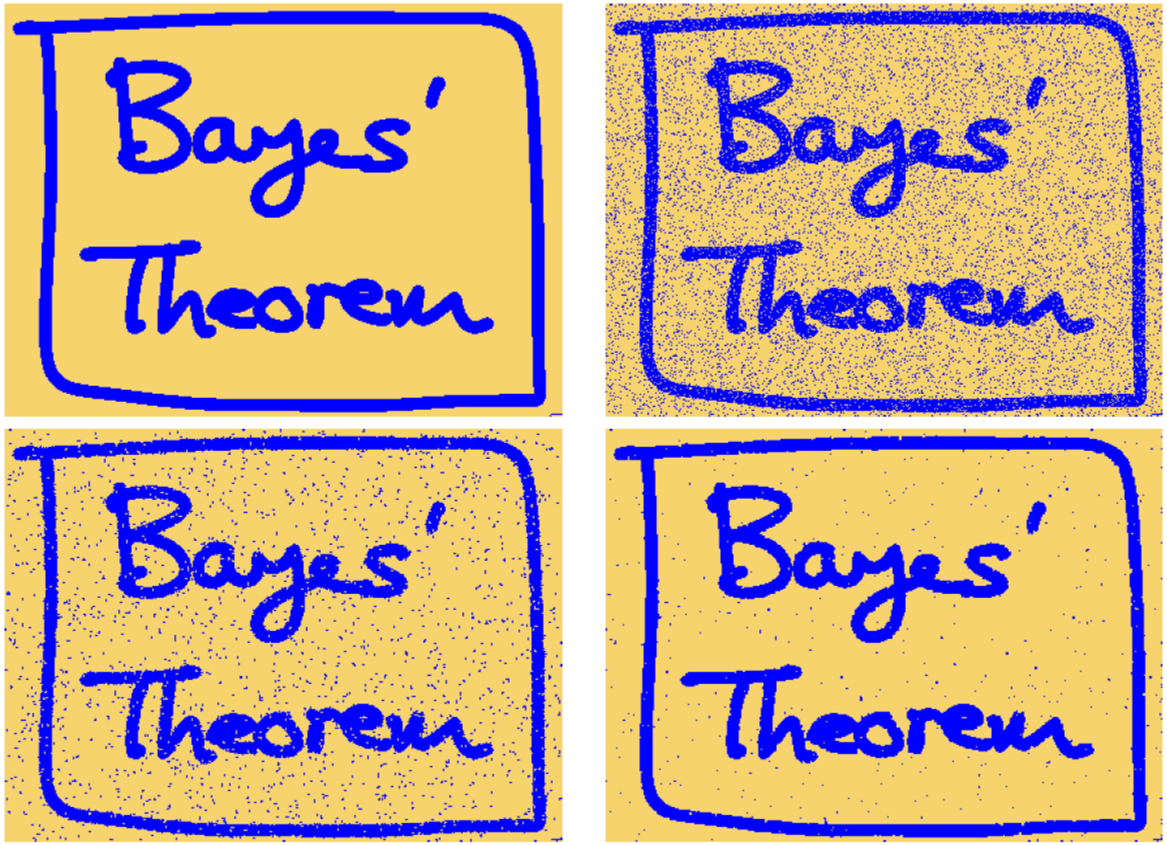
- The quantity *Z,* called the *partition function,* is a normalization constant

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C)$$

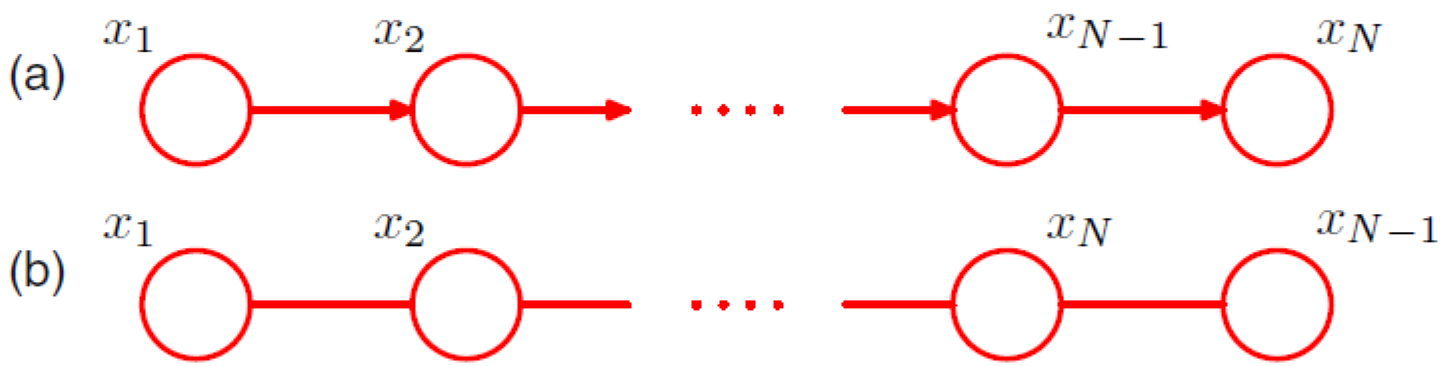
- Potential functions $\psi_C(\boldsymbol{x}_C)$ are **strictly positive.** Possible choice

$$\psi_C(\mathbf{x}_C) = \exp\left\{-E(\mathbf{x}_C)\right\}$$

Image de-noising



Relation to directed graphs



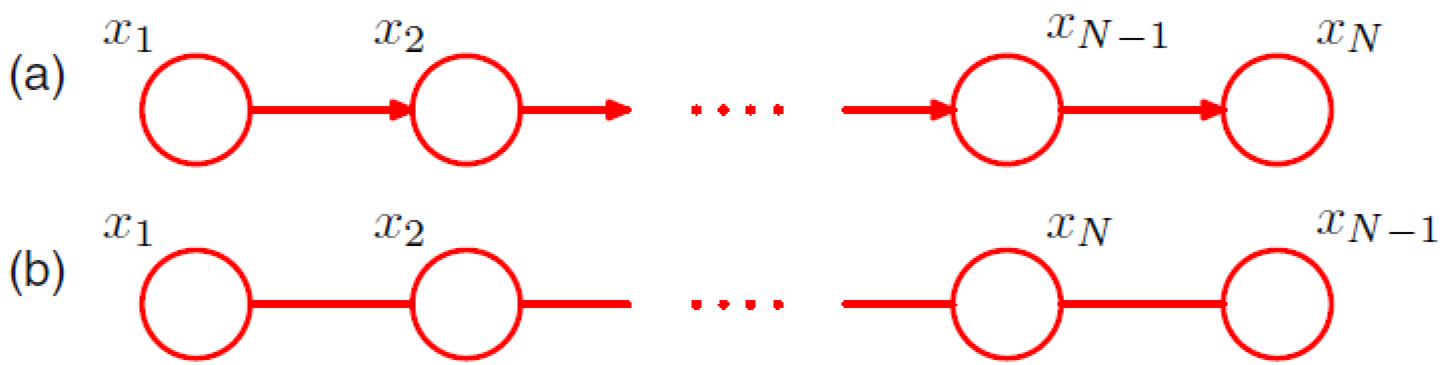
- Joint distribution:
  - Directed:

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2)\cdots p(x_N|x_{N-1})$$

  - Undirected:

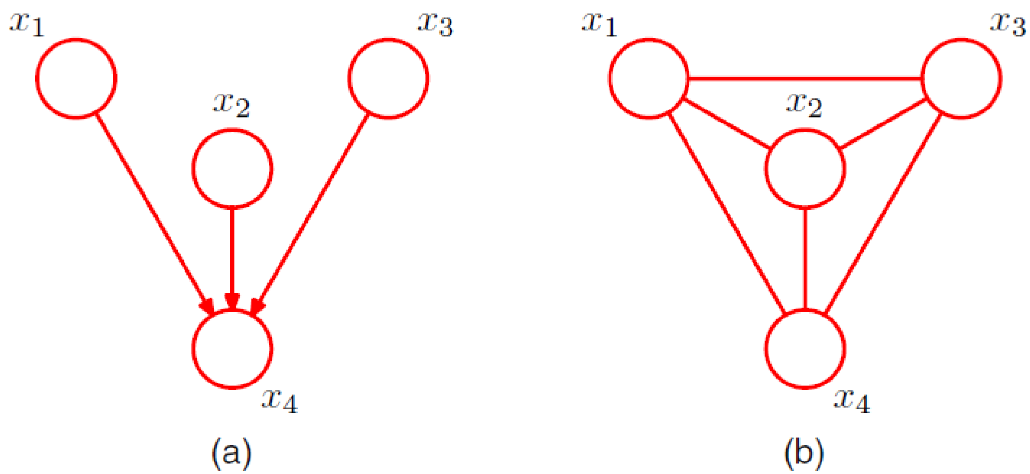
$$p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3)\cdots\psi_{N-1,N}(x_{N-1},x_N)$$

Relation to directed graphs



$$
\begin{aligned}
\psi_{1,2}(x_1, x_2) &= p(x_1)p(x_2|x_1) \\
\psi_{2,3}(x_2, x_3) &= p(x_3|x_2) \\
&\vdots \\
\psi_{N-1,N}(x_{N-1}, x_N) &= p(x_N|x_{N-1})
\end{aligned}
$$

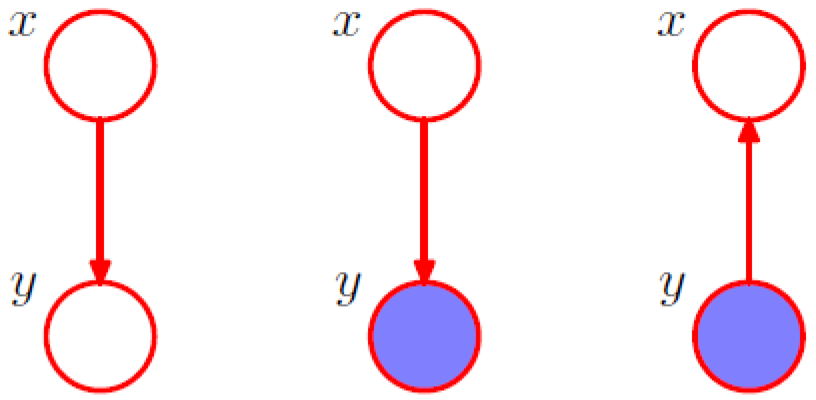
Relation to directed graphs



$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)$$

- this process of ‘marrying the parents’ has become known as *moralization*, and the resulting undirected graph, after dropping the arrows, is called the *moral graph.*

Inference in Graphical Models

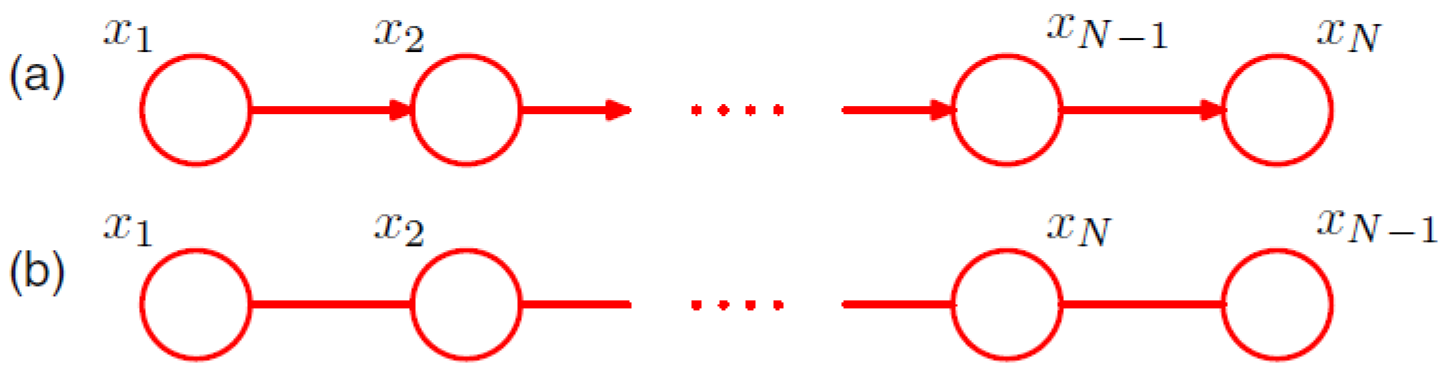


(a) (b) (c)

$$p(x, y) = p(x)p(y|x)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

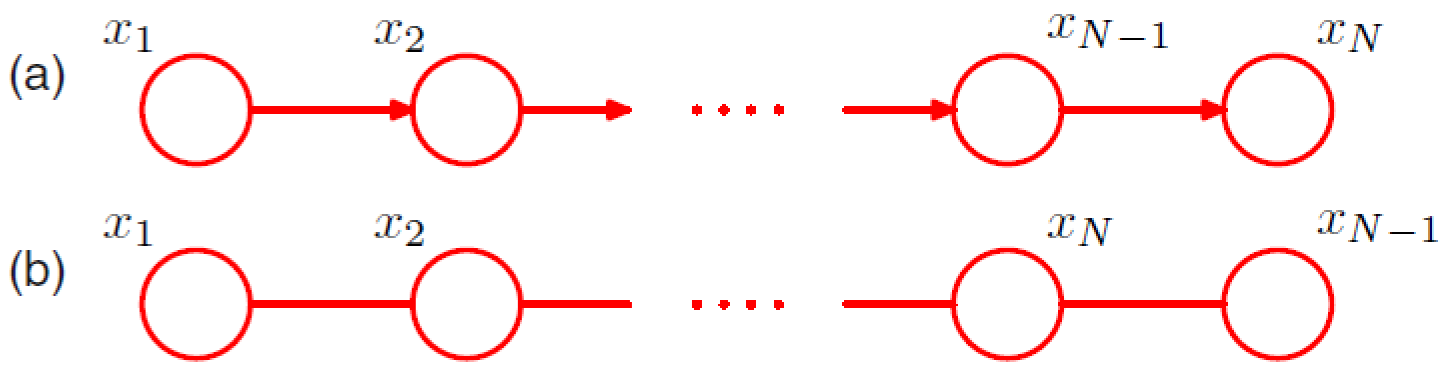
Inference on a chain



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

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

Inference on a chain



$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right]\right] \cdots \right]}_{\mu_\alpha(x_n)} \underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right] \cdots \right]}_{\mu_\beta(x_n)} . \tag{8.52}$$

Inference on a chain

Passing of local *messages* around on the graph

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

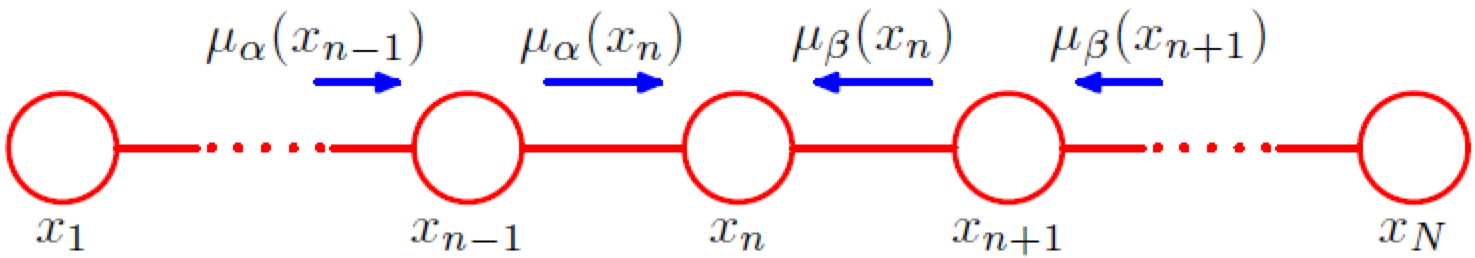
$$\underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots \right]}_{\mu_\beta(x_n)} . \qquad (8.52)$$

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)$$

Inference on a chain

Passing of local *messages* around on the graph



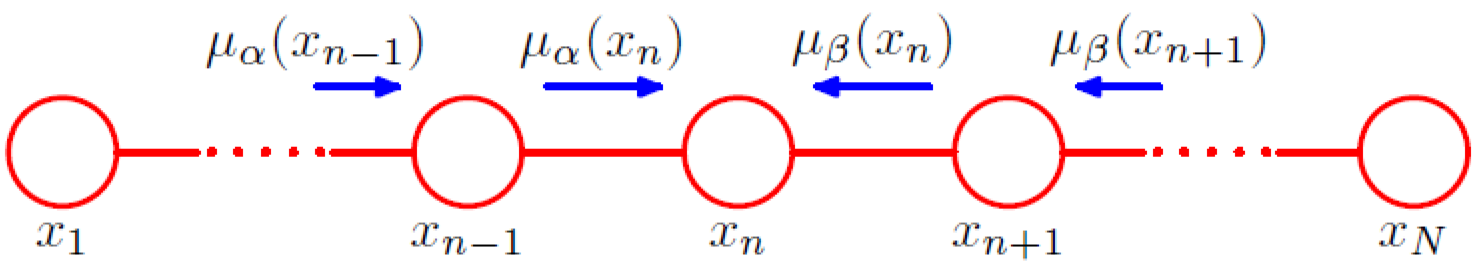
$$\begin{aligned}\mu_\alpha(x_n) &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right] \\ &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}).\end{aligned}$$

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)$$

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

Inference on a chain

Passing of local *messages* around on the graph

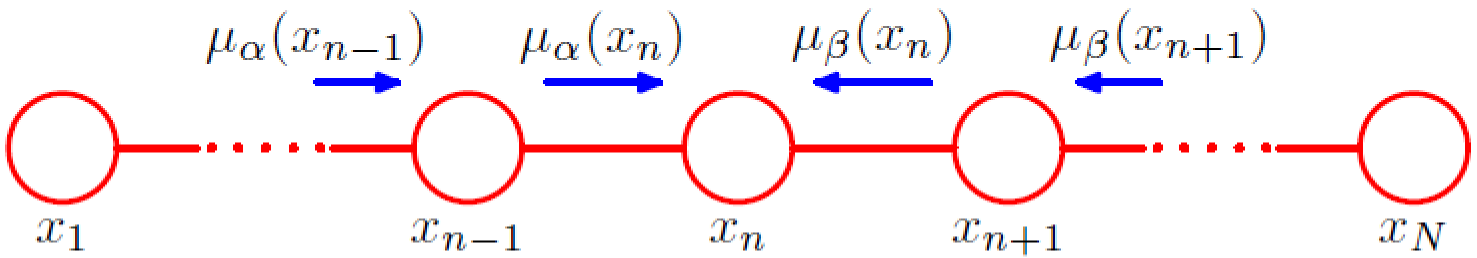


$$\begin{aligned} \mu_\beta(x_n) &= \sum_{x_{n+1}} \psi_{n+1,n}(x_{n+1}, x_n) \left[\sum_{x_{n+2}} \cdots \right] \\ &= \sum_{x_{n+1}} \psi_{n+1,n}(x_{n+1}, x_n) \mu_\beta(x_{n+1}). \end{aligned}$$

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)$$

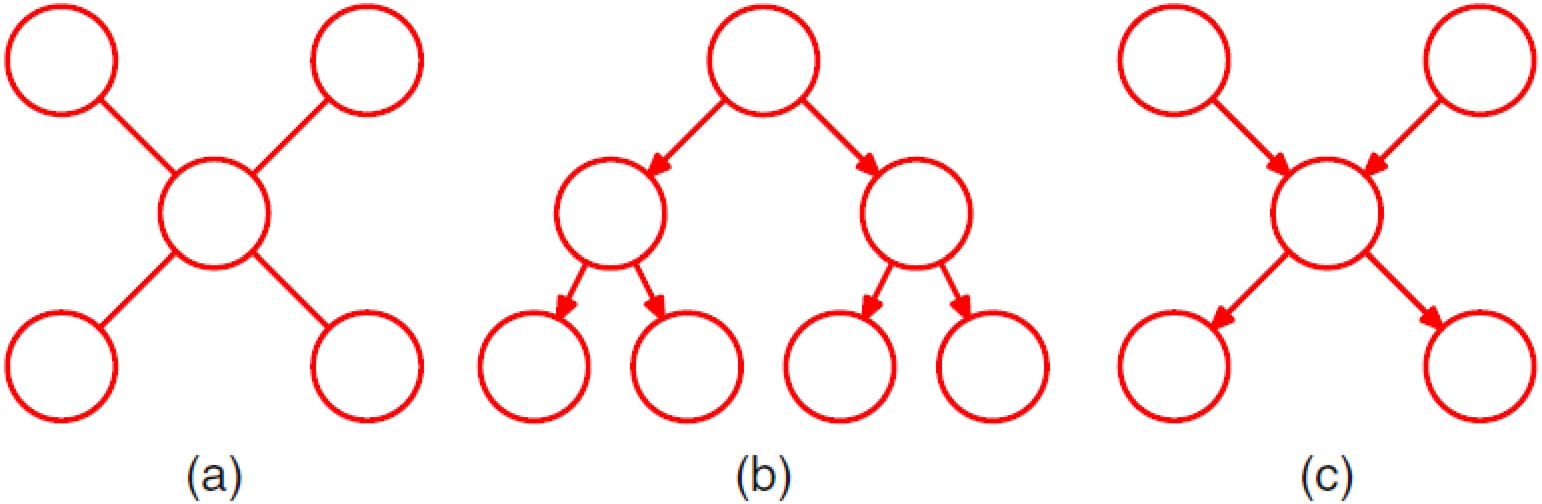
Inference on a chain

Passing of local *messages* around on the graph



$$p(x_{n-1}, x_n) = \frac{1}{Z}\mu_\alpha(x_{n-1})\psi_{n-1,n}(x_{n-1}, x_n)\mu_\beta(x_n)$$

Tree



Factor graph

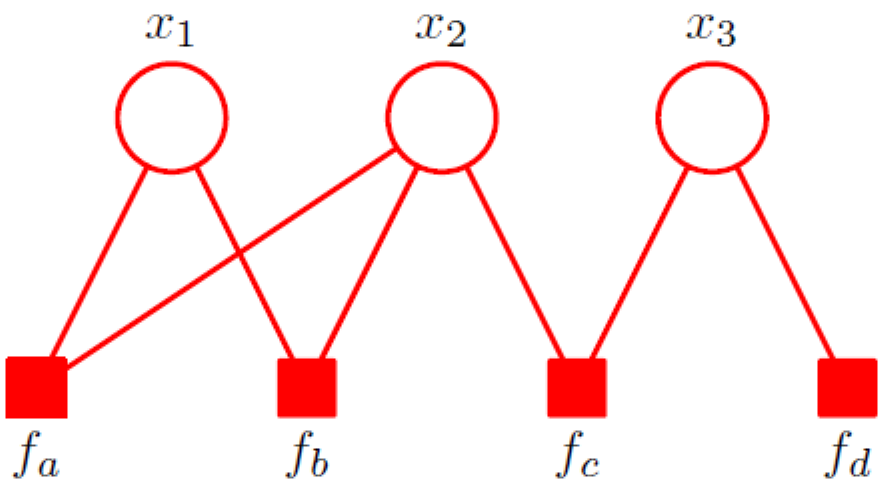
- the joint distribution over a set of variables in the form of a product of factors

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

- where $\mathbf{x}_s$ denotes a subset of the variables

Factor graph

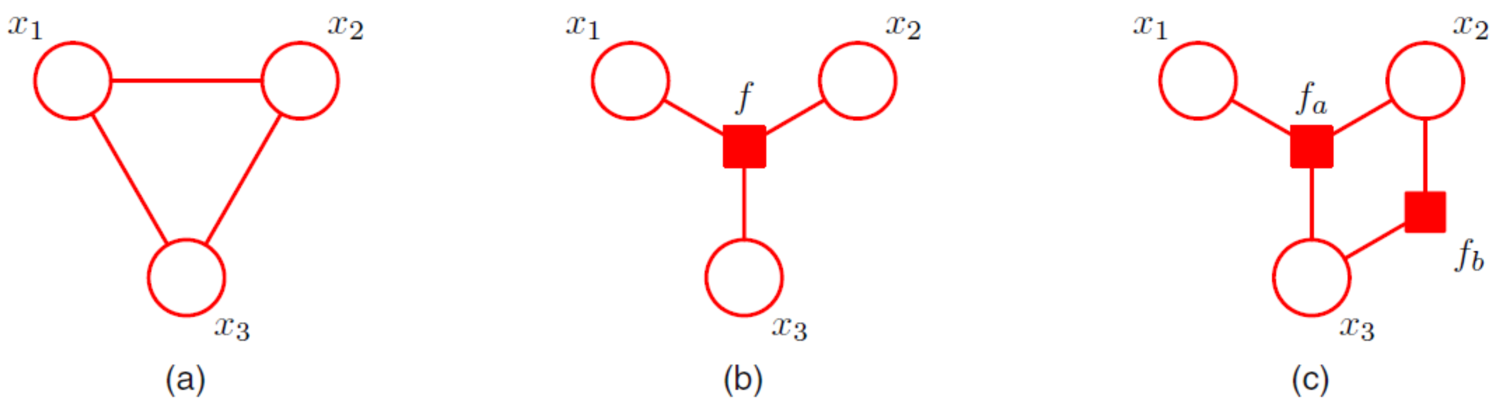
$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3)$$



Factor graph

- an undirected graph => a factor graph
  - create variable nodes corresponding to the nodes in the original undirected graph
  - create additional factor nodes corresponding to the maximal cliques $\mathbf{x}_s$
  - Multiple choices of f_g

Factor graph



(a) An undirected graph with a single clique potential *ψ(x1, x2, x3)*.

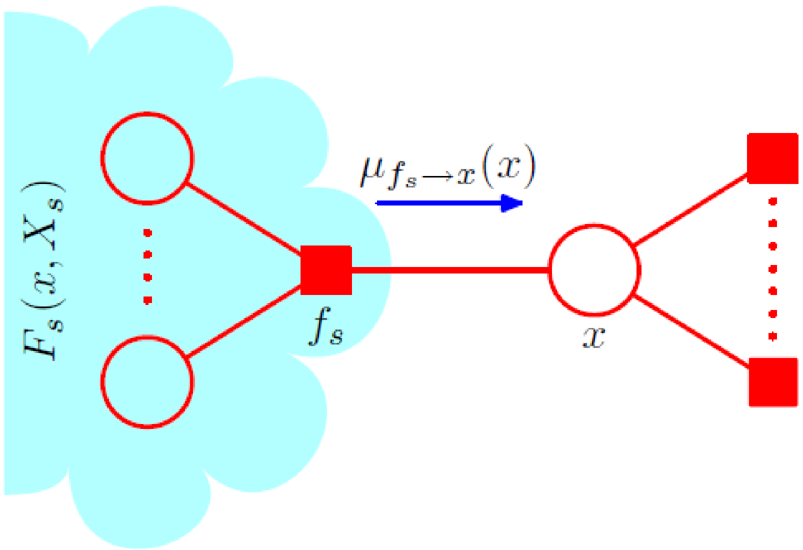
(b) A factor graph with factor *f(x1, x2, x3) = ψ(x1, x2, x3)* representing the same distribution as the undirected graph.

(c) A *different factor* graph representing the same distribution, whose factors satisfy *fa(x1, x2, x3)fb(x1, x2) = ψ(x1, x2, x3)*.

The sum-product algorithm

- The problem of finding the marginal *p(x)* for particular variable node *x*

$$p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x}) \qquad\qquad p(\mathbf{x}) = \prod_{s \in \mathrm{ne}(x)} F_s(x, X_s)$$

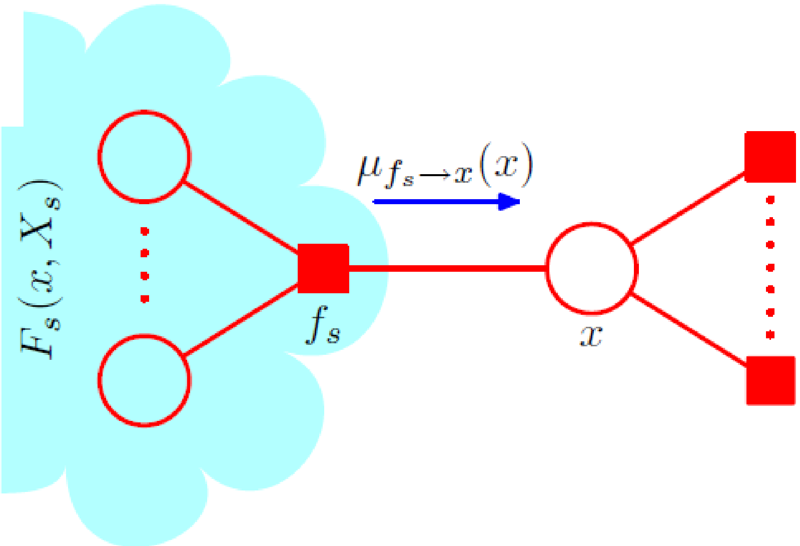


The sum-product algorithm

- The problem of finding the marginal *p(x)* for particular variable node *x*

$$
\begin{aligned}
p(x) &= \prod_{s \in \mathrm{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right] \\
&= \prod_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x).
\end{aligned}
$$

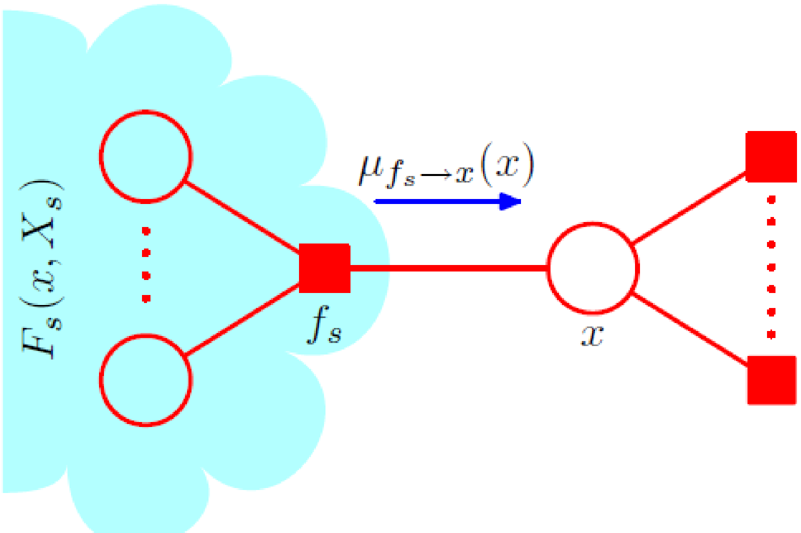
$$
\mu_{f_s \to x}(x) \equiv \sum_{X_s} F_s(x, X_s)
$$



The sum-product algorithm

- The problem of finding the marginal *p(x)* for particular variable node *x*

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

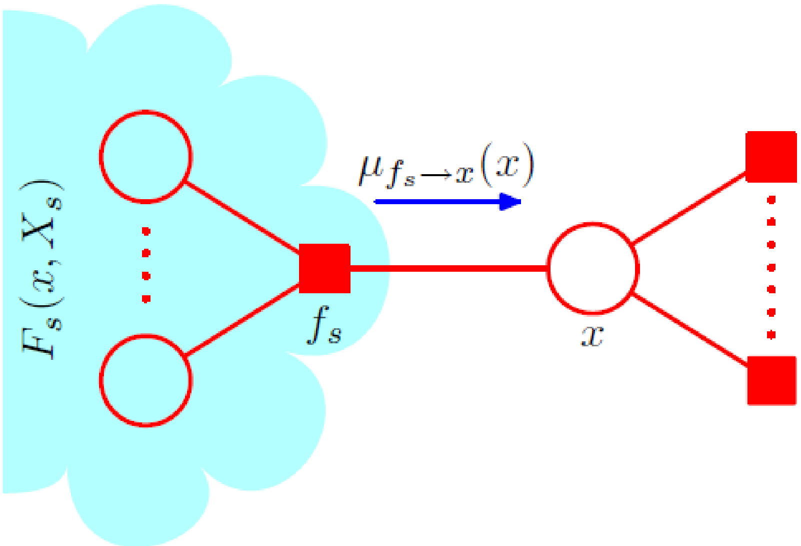


The sum-product algorithm

- The problem of finding the marginal *p(x)* for particular variable node *x*

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

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

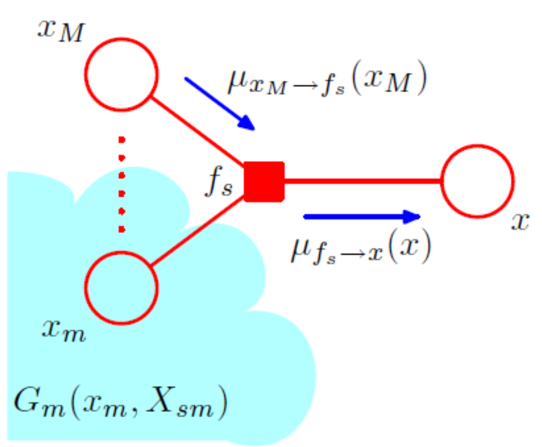


The sum-product algorithm

- The problem of finding the marginal *p(x)* for particular variable node *x*

$$\begin{aligned}\mu_{f_s \to x}(x) &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \left[\sum_{X_{xm}} G_m(x_m, X_{sm}) \right] \\ &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \setminus x} \mu_{x_m \to f_s}(x_m) \qquad (8.66)\end{aligned}$$

$$\mu_{x_m \to f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm})$$



Junction tree algorithm

- deal with graphs having loops
- Algorithm:
  - directed graph => undirected graph (moralization)
  - The graph is triangulated
  - join tree
  - Junction tree
  - a two-stage message passing algorithm, essentially equivalent to the sum-product algorithm

Graph inference example

- Computer-Generated Residential Building Layouts [SIG ASIA 2010]

The End

新浪微博：@浙大张宏鑫

微信公众号：