Graphical Models

Léon Bottou

COS 424 – 4/15/2010

Introduction

People like drawings better than equations

– A graphical model is a diagram representing certain aspects of the algebraic structure of a probabilistic model.

Purposes

– Visualize the structure of a model.
– Investigate conditional independence properties.
– Some computations are more easily expressed on a graph than written as equations with complicated subscripts.

Summary

Summary

I. Directed graphical models
II. Undirected graphical models
III. Inference in graphical models

More

– David Blei runs a complete course on graphical models.

I. Directed graphical models

“Bayesian Networks”

(Pearl 1988)

A pattern for independence assumptions

Probability distribution

$$P(x_1, x_2, x_3, x_4)$$

Bayesian chain theorem

$$P(x_1, x_2, x_3, x_4) = P(x_1)\, P(x_2|x_1)\, P(x_3|x_1, x_2)\, P(x_4|x_1, x_2, x_3)$$

Independence assumptions

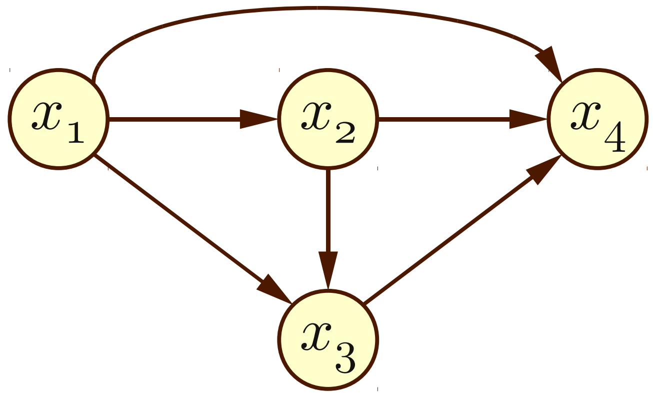
$$\begin{aligned} P(x_1, x_2, x_3, x_4) &= P(x_1)\, P(x_2|x_1)\, P(x_3|x_1, x_2)\, P(x_4|x_1, x_2, x_3) \\ &= P(x_1)\, P(x_2|x_1)\, P(x_3|x_1)\, P(x_4|x_1, x_2) \end{aligned}$$

Graphical representation

Bayesian chain theorem

$$P(x_1, x_2, x_3, x_4) = P(x_1)\, P(x_2|x_1)\, P(x_3|x_1, x_2)\, P(x_4|x_1, x_2, x_3)$$

Directed acyclic graph

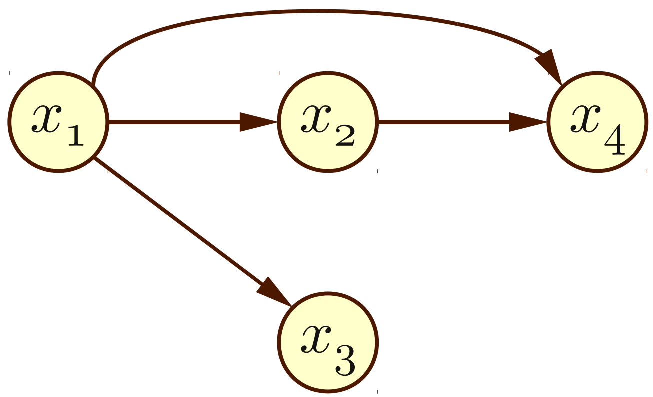


Arrows do not represent causality!

Graphical representation

Independence assumptions

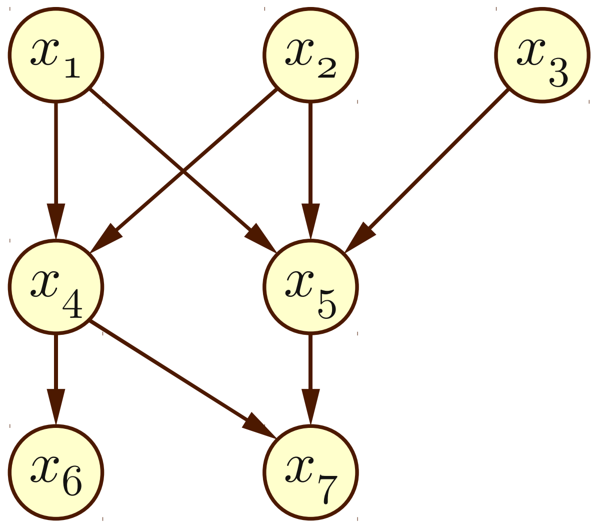
$$\begin{aligned} P(x_1, x_2, x_3, x_4) &= P(x_1)\, P(x_2|x_1)\, P(x_3|x_1, x_2)\, P(x_4|x_1, x_2, x_3) \\ &= P(x_1)\, P(x_2|x_1)\, P(x_3|x_1)\, P(x_4|x_1, x_2) \end{aligned}$$



Missing links represent independence assumptions

A more complicated example

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



Parametrization

The graph says nothing about the parametric form of the probabilities.

– Discrete distributions

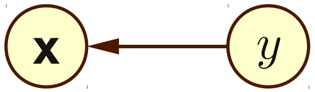
– Continuous distributions

Discrete distributions

Input $\mathbf{x} = (x_1, x_2 \dots x_d) \in \{0,1\}^d$.
Class $y \in \{A_1, \dots, A_k\}$.

General generative model

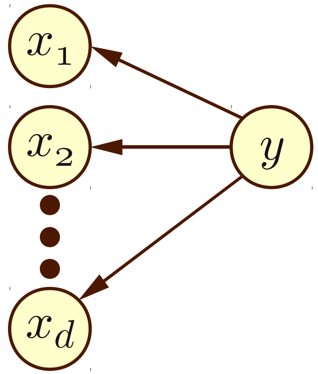
$$P(\mathbf{x}, y) = P(y)\,P(\mathbf{x}|y)$$



– k parameters for $P(y)$
– $k\,2^d$ parameters for $P(\mathbf{x}|y)$

Naïve Bayes model

$$P(\mathbf{x}, y) = P(y)\,P(x_1|y)\,\dots P(x_d|y)$$

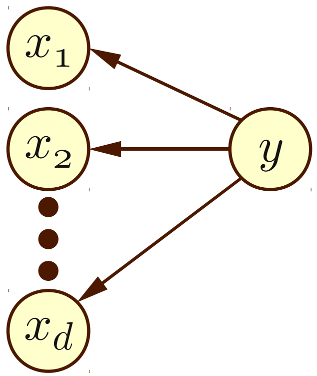


– k parameters for $P(y)$
– $k\,d$ parameters for $P(\mathbf{x}|y)$

Discrete distributions

Naïve Bayes model

$$P(\mathbf{x}, y) = P(y)\, P(x_1|y)\, \ldots P(x_d|y)$$



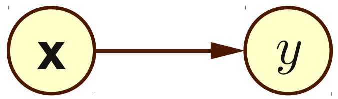
$$\hat{y}(\mathbf{x}) = \arg\max_y P(\mathbf{x}, y)$$

– k parameters for $P(y)$.
– $k\,d$ parameters for $P(\mathbf{x}|y)$.

Fails when the x_i are correlated !

Linear discriminant model

$$P(\mathbf{x}, y) = P(\mathbf{x})\, P(y|\mathbf{x})$$



$$\begin{aligned} \hat{y}(\mathbf{x}) &= \arg\max_y P(\mathbf{x}, y) \\ &= \arg\max_y P(y|\mathbf{x}) \end{aligned}$$

– $k(d+1)$ parameters for $P(y|\mathbf{x})$.
– 2^d *unused* parameters for $P(\mathbf{x})$.

Works when the x_i are correlated !

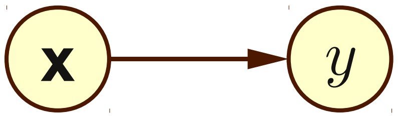
Continuous distributions

Linear regression

– Input $\mathbf{x} = (x_1, x_2 \dots x_d) \in \mathbb{R}^d$.

– Output $y \in \mathbb{R}$.

$$P(\mathbf{x}, y) = P(y|\mathbf{x})\, P(\mathbf{x})$$



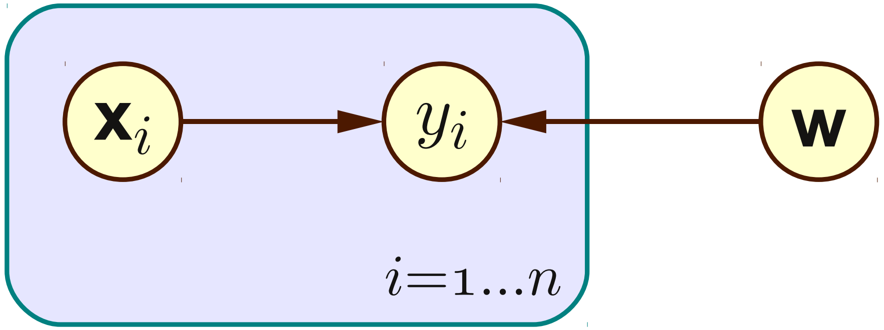
$$P(y|\mathbf{x}) \propto \exp\left(-\frac{1}{2\sigma^2}\left(y - \mathbf{w}^\top \mathbf{x}\right)^2\right)$$

No need to model $P(\mathbf{x})$.

Bayesian regression

Consider a dataset $\mathcal{D} = \{ (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \}$.

$$P(\mathcal{D}, \mathbf{w}) = P(\mathbf{w})\, P(\mathcal{D}|\mathbf{w}) = P(\mathbf{w}) \prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w})\, P(\mathbf{x}_i)$$

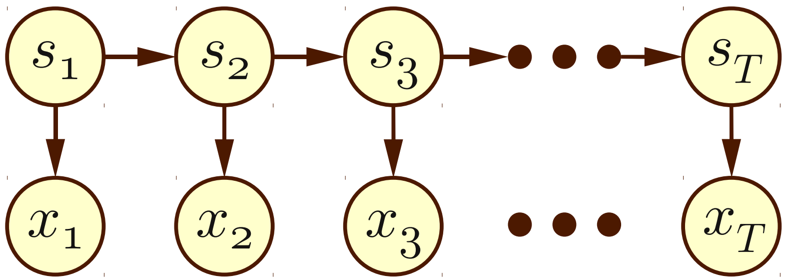


Plates represent repeated subgraphs.

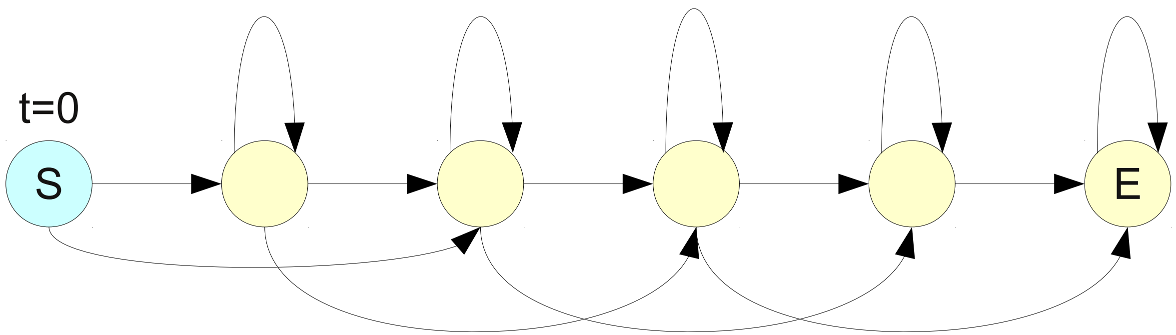
Although the parameter $\mathbf{w}$ is explicit,
other details about the distributions are not.

Hidden Markov Models

$$P(x_1 \dots x_T,\, s_1 \dots s_T) = P(s_1)\, P(x_1|s_1)\, P(s_2|s_1)\, P(x_2|s_2)\, \dots P(s_T|s_{T-1})\, P(x_T|s_T)$$

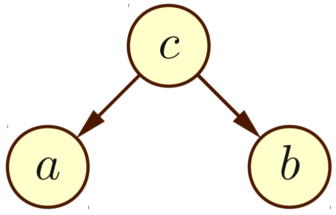


What is the relation between this graph and that graph?



Conditional independence patterns (1)

Tail-to-tail

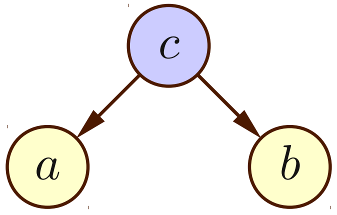


$$P(a,b,c) = P(a|c)\,P(b|c)\,P(c)$$

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

$$\neq P(a)\,P(b) \quad \textit{in general}$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$



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

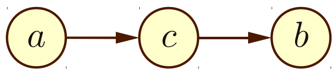
$$P(a,b|c) = P(a,b,c)/P(c)$$

$$= P(a|c)P(b|c)$$

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

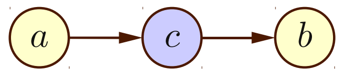
Conditional independence patterns (2)

Head-to-tail



$$
\begin{aligned}
P(a,b,c) &= P(a)\,P(c|a)\,P(b|c) \\
P(a,b) &= \sum_c P(a)\,P(c|a)\,P(b|c) \\
&= P(a)\,\sum_c P(b,c|a) \\
&= P(a)\,P(b|a) \\
&\neq P(a)\,P(b) \quad \textit{in general}
\end{aligned}
$$

$$\boldsymbol{a \not\perp\!\!\!\perp b \mid \emptyset}$$

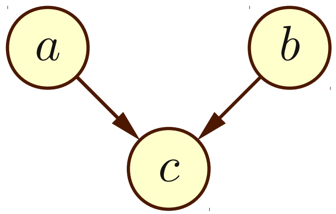


$$
\begin{aligned}
P(a,b,c) &= P(a)\,P(c|a)\,P(b|c) \\
&= P(a,c)\,P(b|c) \\
P(a,b|c) &= P(a,b,c)/P(c) \\
&= P(a|c)P(b|c)
\end{aligned}
$$

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

Conditional independence patterns (3)

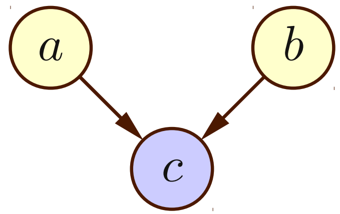
Head-to-head



$$P(a,b,c) \;=\; P(a)\,P(b)\,P(c|a,b)$$

$$P(a,b) \;=\; \sum_c P(a)\,P(b)\,P(c|a,b)$$
$$\;=\; P(a)\,P(b)\,\sum_c P(c|a,b))$$
$$\;=\; P(a)\,P(b)$$

$$\boldsymbol{a \perp\!\!\!\perp b \mid \emptyset}$$



$$P(a,b,c) \;=\; P(a)\,P(b)\,P(c|a,b)$$

$P(a,b|c) \;\neq\; P(a|c)P(b|c)$ *in general*

Example:

$c =$ *"the house is shaking"*

$a =$ *"there is an earthquake"*

$b =$ *"a truck hits the house"*

$$\boldsymbol{a \not\perp\!\!\!\perp b \mid c}$$

D-separation

Problem

– Consider three disjoint sets of nodes: A, B, C.

– When do we have $A \perp\!\!\!\perp B \mid C$?

Definition

A and B are *d-separated* by C if all paths from $a \in A$ to $b \in B$

– contain a head-to-tail or tail-to-tail node $c \in C$, or

– contain a head-to-head node c such that neither c nor any of its descendants belongs to C.

Theorem

A and B are *d-separated* by C $\iff$ $A \perp\!\!\!\perp B \mid C$

II. Undirected graphical models

"Markov Random Fields"

Another independence assumption pattern

Boltzmann distribution

$$P(\mathbf{x}) = \frac{1}{Z} \exp\big(-E(\mathbf{x})\big) \quad \text{with} \quad Z = \sum_{\mathbf{x}} \exp\big(-E(\mathbf{x})\big)$$

– The function $E(\mathbf{x})$ is called *energy function*.
– The quantity Z is called the *partition function*.

Markov Random Field

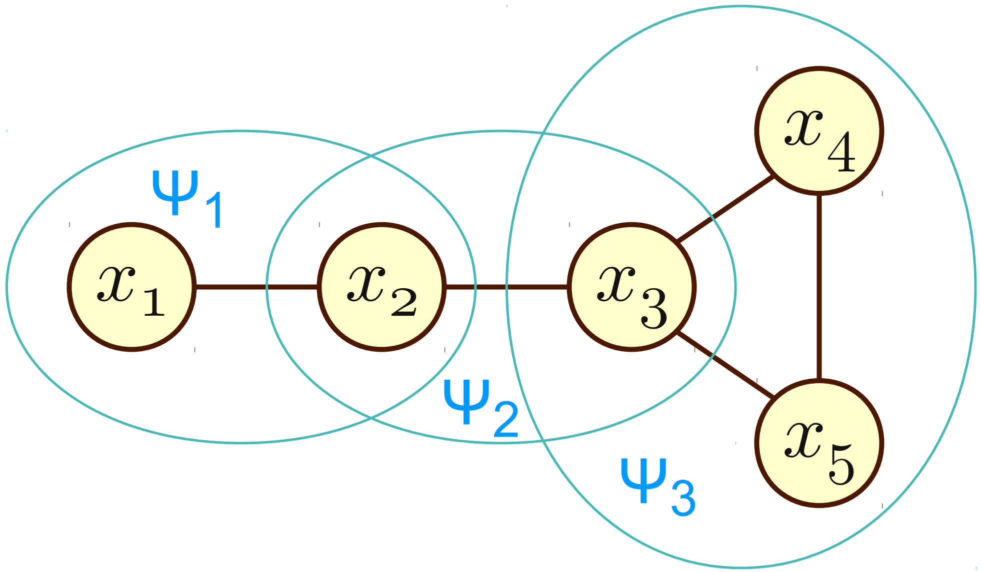
– Let $\{\mathbf{x}_C\}$ be a family of subsets of the variables $\mathbf{x}$.
– The distribution $P(\mathbf{x})$ is a *Markov Random Field* with cliques $\{\mathbf{x}_C\}$ if there are functions $E_C(\mathbf{x}_C)$ such that $E(\mathbf{x}) = \sum_C E_C(\mathbf{x}_C)$.

Equivalently,

$$P(\mathbf{x}) = \frac{1}{Z} \prod_C \Psi_C(\mathbf{x}_C) \quad \text{with} \quad \Psi_C(\mathbf{x}_C) = \exp(-E_C(\mathbf{x}_C)) > 0\,.$$

Graphical representation

$$P(x_1, x_2, x_3, x_4, x_5) = \frac{1}{Z}\,\Psi_1(x_1, x_2)\,\Psi_2(x_2, x_3)\,\Psi_3(x_3, x_4, x_5)$$



– Completely connect the nodes belonging to each $\mathbf{x}_C$.
– Each subset $\mathbf{x}_C$ forms a *clique* of the graph.

Markov Blanket

Definition

– The Markov blanket of x is the minimal subset of variables $\mathcal{B}_x$ of the variables $\mathbf{x}$ such that $P(x \,|\, \mathbf{x} \setminus x) = P(x \,|\, \mathcal{B}_x)$.

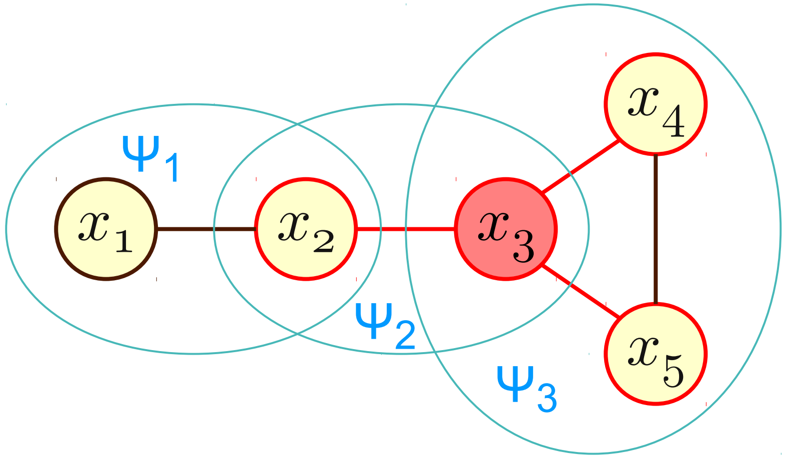
Example

$$
\begin{aligned}
P(x_3 \,|\, x_1, x_2, x_4, x_5) &= \frac{\Psi_1(x_1, x_2)\, \Psi_2(x_2, x_3)\, \Psi_3(x_3, x_4, x_5)}{\sum_{x_3'} \Psi_1(x_1, x_2)\, \Psi_2(x_2, x_3')\, \Psi_3(x_3', x_4, x_5)} \\
&= \frac{\Psi_2(x_2, x_3)\, \Psi_3(x_3, x_4, x_5)}{\sum_{x_3'} \Psi_2(x_2, x_3')\, \Psi_3(x_3', x_4, x_5)} \\
&= P(x_3 \,|\, x_2, x_4, x_5)
\end{aligned}
$$

Graph and Markov blanket

The Markov blanket of a MRF variable is the set of its neighbors.

$$P(x_3 \,|\, x_1, x_2, x_4, x_5) = P(x_3 \,|\, x_2, x_4, x_5)$$



Consequence

– Consider three disjoint sets of nodes: A, B, C.

$$A \perp\!\!\!\perp B \mid C \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} \text{Any path between } a \in A \text{ and } b \in B \\ \text{passes through a node } c \in C. \end{array} \right.$$

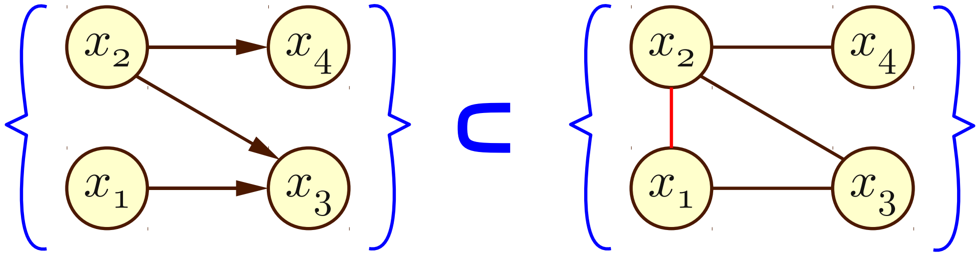
Conversely (Hammersley-Clifford theorem)

– Any distribution that satisfies such properties with respect to an undirected graph is a Markov Random Field.

Directed vs. undirected graphs

Consider a directed graph.

$$P(\mathbf{x}) = \underbrace{P(x_1)}_{\Psi_1(x_1)} \; \underbrace{P(x_2)}_{\Psi_2(x_2)} \; \underbrace{P(x_3|x_1,x_2)}_{\Psi_3(x_1,x_2,x_3)} \; \underbrace{P(x_4|x_2)}_{\Psi_4(x_2,x_4)} \qquad (Z=1)$$



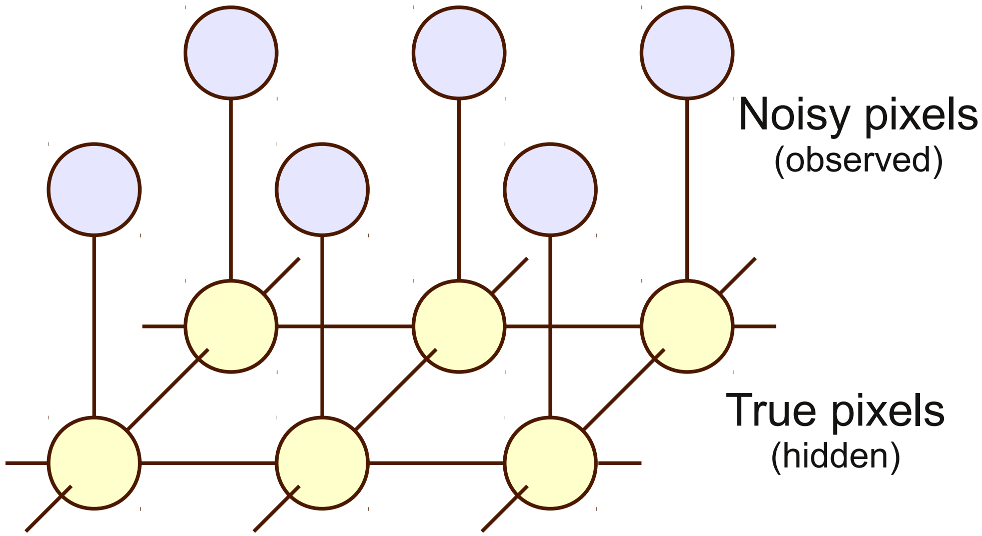
The opposite inclusion is not true because the undirected graph marries the parents of x_3 with a moralization link.

Directed and undirected graphs represent different sets of distributions. Neither set is included in the other one.

Example: image denoising

Noise model: randomly flipping a small proportion of the pixels.

Image model: pixel distribution given its four neighbors.



Inference problem

– Given the observed noisy pixels,
reconstruct the true pixel distributions.

III. Inference in graphical models

Inference

Partition the variables

– A: the variables of interest.

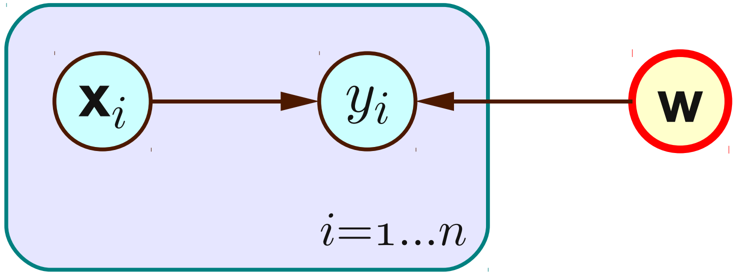
– B: the observed variables.

– R: the rest.

We want $P(A|B)$

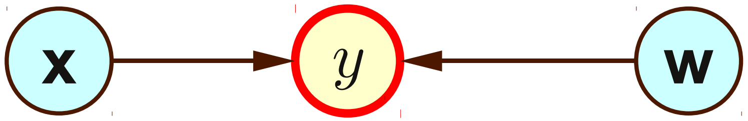
Inference

Inference for learning



$P(\mathbf{w} \mid \mathcal{D})$?

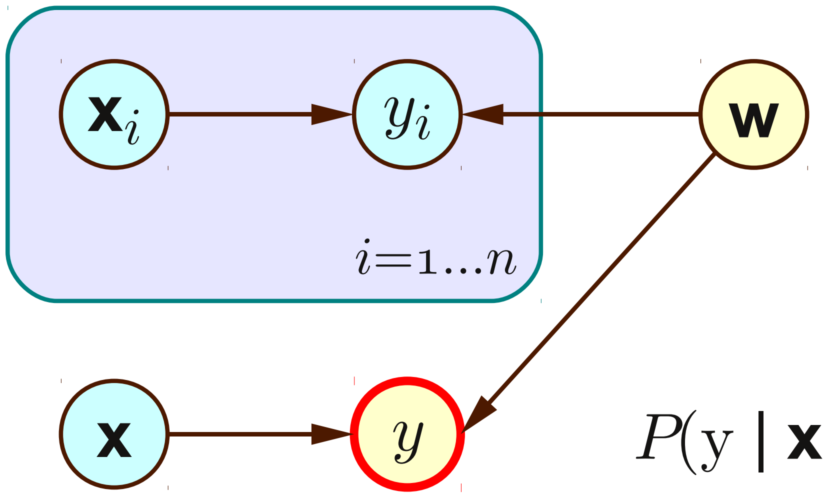
Inference for recognition



$P(\mathrm{y} \mid \mathbf{x}, \mathbf{w})$?

Inference

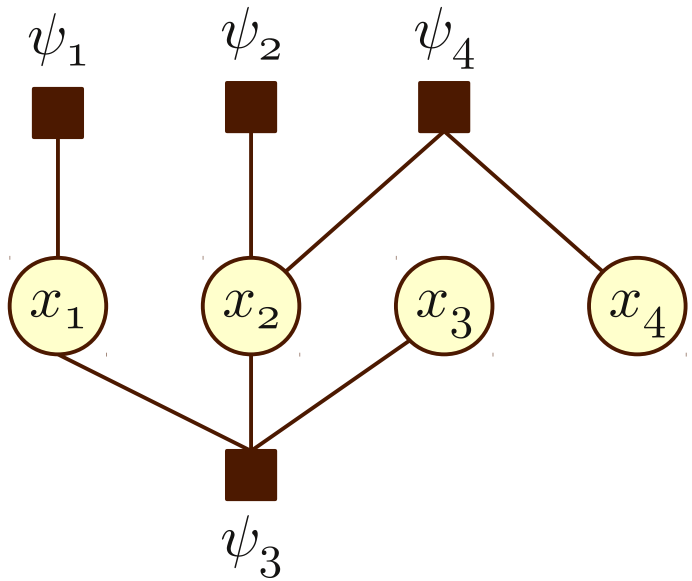
Inference for both (Bayesian averaging)



$P(\mathrm{y} \mid \mathbf{x}, \mathcal{D})$?

Factor graph

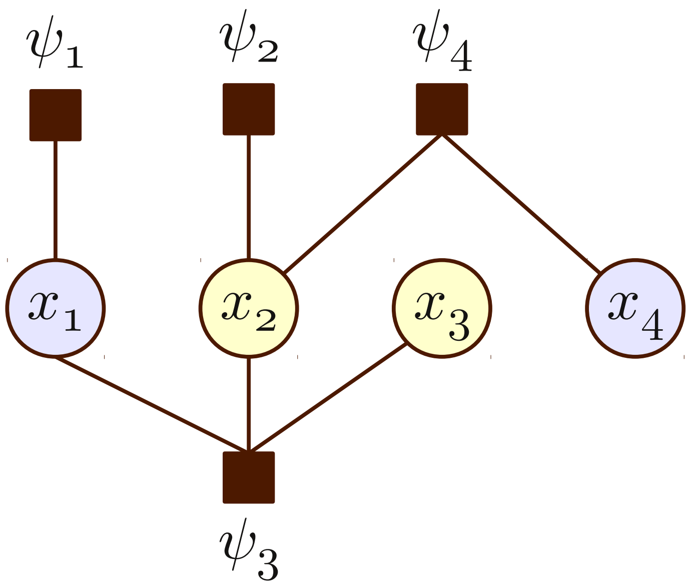
$$P(\mathbf{x}) \propto \Psi_1(x_1)\,\Psi_2(x_2)\,\Psi_3(x_1,x_2,x_3)\,\Psi_4(x_2,x_4)$$



A factor graph is a bipartite undirected graph.

Gibbs sampling

A computationally intensive inference algorithm



Clamp the observed variables.
Randomly initialize the other variables.

Repeat:

– Pick one unobserved variable x.
– Compute $P(\, x \,|\, \mathrm{ne}(\mathrm{ne}(x)) \,)$.
– Pick a new value for x accordingly.

Observe the empirical distribution
of the variables of interest.

Direct computation

Sum-Product algorithm

The sum-product algorithm efficiently solves the problem
when the factor graph (restricted to the unobserved variables) is a tree.

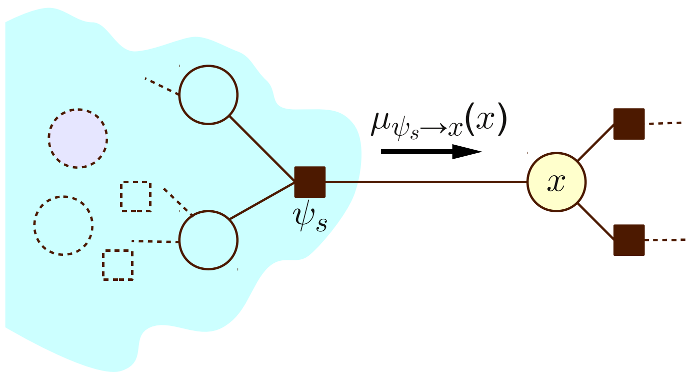
– directed graphical models: trees, polytrees, . . .
– undirected graphical models: trees, and more . . .

Particular cases

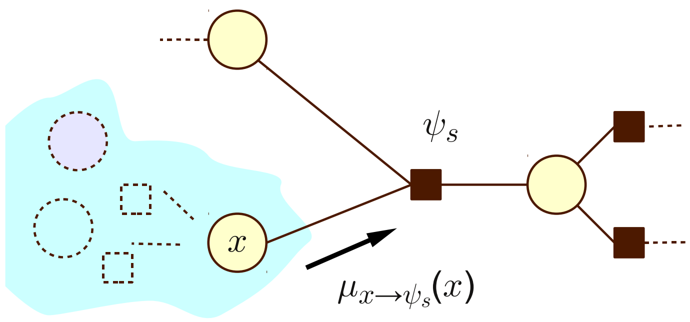
– Forward algorithm for HMMs.
– Belief propagation for directed graphical models.

Sum-product algorithm (1)

Definitions



$$\mu_{\Psi_s \to x}(x) = \sum_{\mathbf{x}} \prod_{\Psi_C} \Psi_C(\mathbf{x}_C)$$

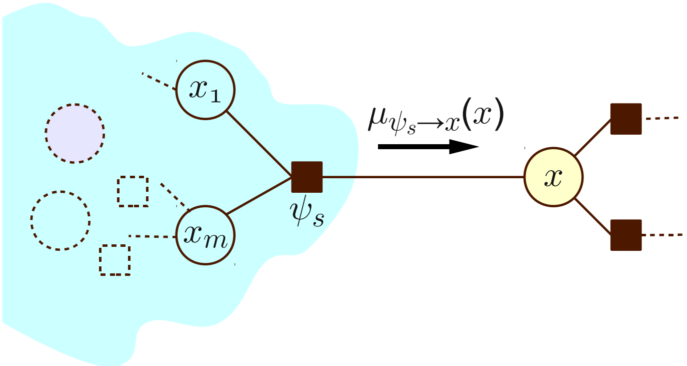


$$\mu_{x \to \Psi_s}(x) = \sum_{\mathbf{x}} \prod_{\Psi_C} \Psi_C(\mathbf{x}_C)$$

– $\mathbf{x}$ represents all unobserved variables other than x in the cyan zone.
– Ψ_C represents all factors in the cyan zone.

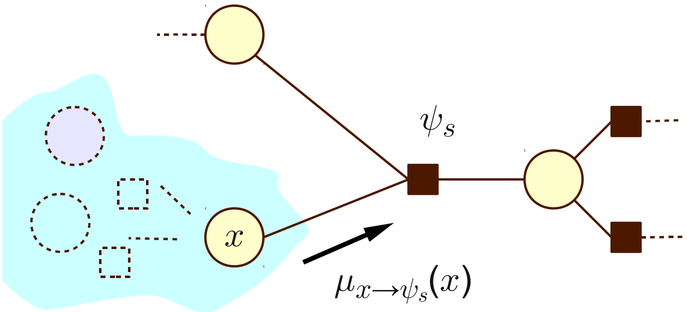
Sum-product algorithm (2)

Recursions



$$\mu_{\Psi_s \to x}(x) = \sum_{x_1..x_m..x_M} \Psi_s(\mathbf{x}_s) \prod_m \mu_{x_m \to \Psi_s}(x_m)$$

$\mu_{\Psi_s \to x}(x) = \Psi_s(x)$ if Ψ_s is a leaf.



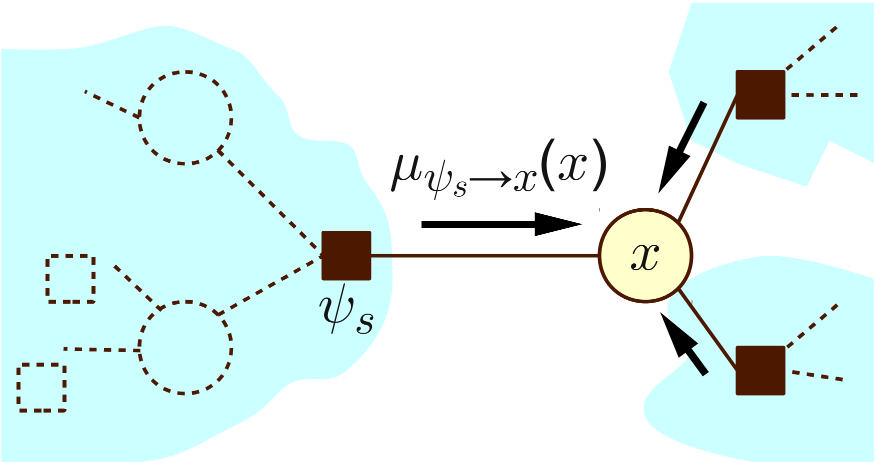
$$\mu_{x \to \Psi_s}(x) = \prod_{l \in \text{ne}(x) \setminus s} \mu_{\Psi_l \to x}(x)$$

$\mu_{x \to \Psi_s}(x) = 1$ if x is a leaf.

– These recursion work because we assume the factor graph is a tree.
– Starting from the leafs, compute the messages μ everywhere.

Sum-product algorithm (3)

Conclusion



$$\tilde{p}(x) = \prod_{s \in \text{ne}(x)} \mu_{\Psi_s \to x}(x)$$

$$P(x) = \frac{\tilde{p}(x)}{\sum_{x'} \tilde{p}(x')}$$

Issues

– Normalization is easy when x is discrete.
  When x is continuous...

– Multiplying all these small numbers causes numerical problems.
  Renormalizing or using logarithms is often necessary.
  This is also true in HMMs.

Max-product

Semi-ring	Algorithm
$\{\, \mathbb{R}^+,\ +,\ \times \,\}$	Sum-product
$\{\, \mathbb{R},\ \oplus,\ + \,\}$	?
$\{\, \mathbb{R}^+,\ \max,\ \times \,\}$	Max-product
$\{\, \mathbb{R},\ \max,\ +\}$	Sum-product

The max-product and max-sum algorithms can be used to compute the most likely values of the hidden variables.

Backtracking requires attention.

Loopy graphs

Junction tree algorithm

– Performs inference in general graphs.
– Quickly becomes intractable.

Graph partitionning algorithms

– Very useful for image segmentation and image processing.
– Only works for certain graphs.

Approximations

– There are coarse approximations.
– There are refined approximations.
– Instead of defining a probabilistic model and approximating, one could work directly with the approximation...

Conclusion

Is it really easier with graphs?

Benefits

– Visualization of the structure.
– Visualization of independence assumptions.
– Elegant generic algorithms for everything.

Drawbacks

– Visualization is incomplete.
– Confusion between directed models and causality.
– The computational cost of normalization is a recurrent issue.
– One has to rederive the algorithms by hand anyway.
– Algorithms for loopy graphs are usually intractable.