Probabilistic Graphical Models (1): representation

Qinfeng (Javen) Shi

The Australian Centre for Visual Technologies,
The University of Adelaide, Australia

15 April 2011

Course Outline

Probabilistic Graphical Models:

1. Representation (Today)
2. Inference
3. Learning
4. Sampling-based approximate inference
5. Temporal models
6. ...

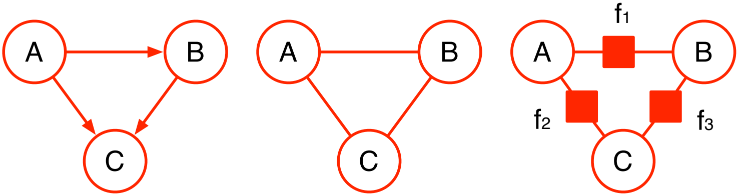
History

- Gibbs (1902) used undirected graphs in particles
- Wright (1921,1934) used directed graph in genetics
- In economists and social sci (Wold 1954, Blalock, Jr. 1971)
- In statistics (Bartlett 1935, Vorobev 1962, Goodman 1970, Haberman 1974)
- In AI, expert system (Bombal *et al.* 1972, Gorry and Barnett 1968, Warner *et al.* 1961)
- Widely accepted in late 1980s. Prob Reasoning in Intelli Sys (Pearl 1988), Pathfinder expert system (Heckerman *et al.* 1992)
- Hot since 2001. CRFs (Lafferty *et al.* 2001), SVM struct (Tsochantaridis etal 2004), M^3Net (Taskar *et al.* 2004), DeepBeliefNet (Hinton *et al.* 2006)

Good books

- Chris Bishop's book "Pattern Recognition and Machine Learning" (Graphical Models are in chapter 8, which is available from his webpage) $\approx$ 60 pages
- Koller and Friedman's "Probabilistic Graphical Models" $>$ 1000 pages
- Stephen Lauritzen's "Graphical Models"
- Michael Jordan's unpublished book "An Introduction to Probabilistic Graphical Models"
- $\ldots$

Three main types of graphical models



(a) Directed graph (b) Undirected graph (c) Factor graph

- Nodes represent random variables.
- Edges represent dependencies between variables
- Factors explicitly show which variables are used in each factor *i.e.* $f_1(A, B)f_2(A, C)f_3(B, C)$

Benefits of graphical models

- Relationships (and interactions) between variables are intuitive (such as conditional independences)
- compactly represent distributions of variables.
- have general inference algorithms (such as message-passing algorithms) to efficiently query $P(A|B = b, C = c)$ or compute $\mathbb{E}_P[f]$ without enumerating all possible values of variables.

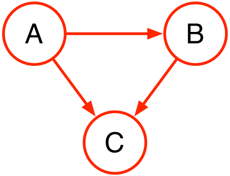
Independences and factorisation

Independences give factorisation.

- Independence
  $A \perp\!\!\!\perp B \Leftrightarrow P(A, B) = P(A)P(B)$
- Conditional Independence
  $A \perp\!\!\!\perp B|C \Leftrightarrow P(A, B|C) = P(A|C)P(B|C)$

From graphs to factorisation

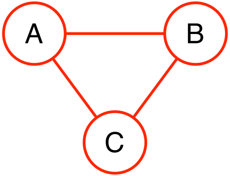
Directed Acyclic Graph:
$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Pa_{x_i})$



$\Rightarrow P(A, B, C) = P(A)P(B|A)P(C|A, B)$

From graphs to factorisation

Undirected Graph:
$P(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{X}_c)$, $Z = \sum_{\mathbf{X}} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{X}_c)$,
where c is an index set of a clique (fully connected subgraph), $\mathbf{X}_c$ is the set of variables indicated by c.

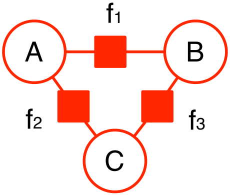


$\Rightarrow P(A, B, C) = \frac{1}{Z} \psi_{c_1}(A, B) \psi_{c_2}(A, C) \psi_{c_3}(B, C)$, when
$\mathbf{X}_{c_1} = \{A, B\}, \mathbf{X}_{c_2} = \{A, C\}, \mathbf{X}_{c_3} = \{B, C\}$
or $P(A, B, C) = \frac{1}{Z} \psi_c(A, B, C)$, when $\mathbf{X}_c = \{A, B, C\}$

From graphs to factorisation

Factor Graph:

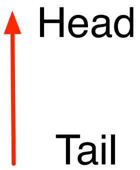
$P(x_1, \ldots, x_n) = \frac{1}{Z}\prod_i f_i(\mathbf{X}_i)$, $Z = \sum_{\mathbf{X}} \prod_i f(\mathbf{X}_i)$

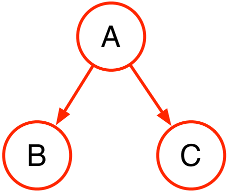


$\Rightarrow P(A, B, C) = \frac{1}{Z} f_1(A, B) f_2(A, C) f_3(B, C)$

From graphs to independences

Case 1: A is said to be tail-to-tail.

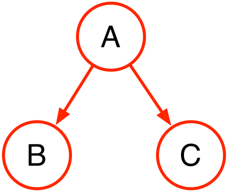




Question: $B \perp\!\!\!\perp C$?

From graphs to independences

Case 1:

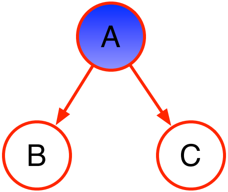


Question: $B \perp\!\!\!\perp C$?
Answer: No.

$$
\begin{aligned}
P(B, C) &= \sum_A P(A, B, C) \\
&= \sum_A P(B|A)P(C|A)P(A) \\
&\neq P(B)P(C) \text{ in general}
\end{aligned}
$$

From graphs to independences

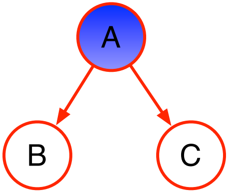
Case 1:



Question: $B \perp\!\!\!\perp C | A$?

From graphs to independences

Case 1:

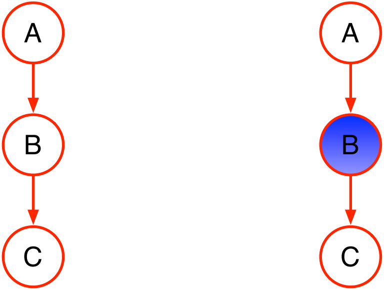


Question: $B \perp\!\!\!\perp C | A$?
Answer: Yes.

$$P(B, C|A) = \frac{P(A, B, C)}{P(A)}$$
$$= \frac{P(B|A)P(C|A)P(A)}{P(A)}$$
$$= P(B|A)P(C|A)$$

From graphs to independences

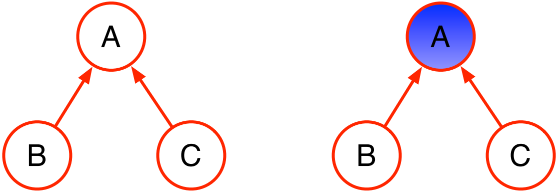
Case 2: B is said to be head-to-tail.



Question: $A \perp\!\!\!\perp C$, $A \perp\!\!\!\perp C | B$?

From graphs to independences

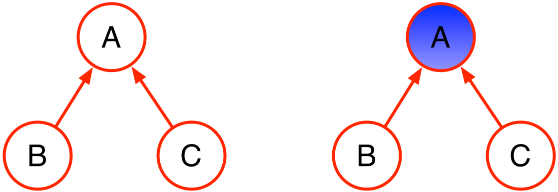
Case 3: A is said to be head-to-head.



Question: $B \perp\!\!\!\perp C$, $B \perp\!\!\!\perp C | A$?

From graphs to independences

Case 3:



Question: $B \perp\!\!\!\perp C$, $B \perp\!\!\!\perp C | A$?

$$\because P(A, B, C) = P(B)P(C)P(A|B, C),$$

$$\therefore P(B, C) = \sum_A P(A, B, C)$$

$$= \sum_A P(B)P(C)P(A|B, C)$$

$$= P(B)P(C)$$

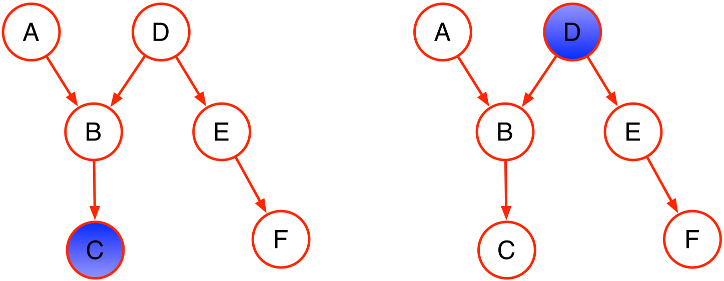
D-separation - def

Graph $G(\mathcal{V}, \mathcal{E})$ and nonintersecting sets $X, Y, O \subset \mathcal{V}$.
How to check $X \perp\!\!\!\perp Y | O$ just by reading the graph G?
Consider all paths from any node $\in X$ to any node $\in Y$. A path is said to be blocked by O, if it includes a node such that either

- exists a node $\in O$ is either head-to-tail or tail-to-tail.
- does not exist a head-to-head node $\in O$, nor any of its descendants $\in O$.

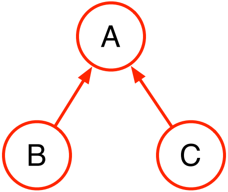
If all paths from X to Y are blocked by O, then X is said to be d-separated (directed separated) from Y by O .

D-separation - Example



Questions:
Is $A \perp\!\!\!\perp F | C$? Check if A is d-separated from F by C.
Is $A \perp\!\!\!\perp F | D$? Check if A is d-separated from F by D.

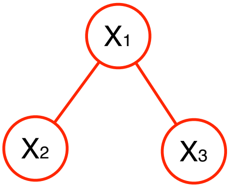
Inference - variable elimination



What is $P(A)$, or $\text{argmax}_{A,B,C} P(A, B, C)$?

$$
\begin{aligned}
P(A) &= \sum_{B,C} P(B)P(C)P(A|B, C) \\
&= \sum_{B} P(B) \sum_{C} P(C)P(A|B, C) \\
&= \sum_{B} P(B)m_1(A, B) \quad (C \text{ eliminated}) \\
&= m_2(A) \quad (B \text{ eliminated})
\end{aligned}
$$

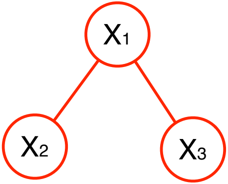
Inference - variable elimination



$$P(x_1, x_2, x_3) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3)$$

$$\begin{aligned}
P(x_1) &= \frac{1}{Z}\sum_{x_2, x_3}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_1)\psi(x_2)\psi(x_3) \\
&= \frac{1}{Z}\psi(x_1)\sum_{x_2}\Big(\psi(x_1, x_2)\psi(x_2)\Big)\sum_{x_3}\Big(\psi(x_1, x_3)\psi(x_3)\Big) \\
&= \frac{1}{Z}\psi(x_1)m_{2\to 1}(x_1)m_{3\to 1}(x_1)
\end{aligned}$$

Inference - variable elimination



$$P(x_2) = \frac{1}{Z}\psi(x_2)\sum_{x_1}\left(\psi(x_1, x_2)\psi(x_1)\sum_{x_3}[\psi(x_1, x_3)\psi(x_3)]\right)$$

$$= \frac{1}{Z}\psi(x_2)\sum_{x_1}\psi(x_1, x_2)\psi(x_1)m_{3\to 1}(x_1)$$

$$= \frac{1}{Z}\psi(x_2)m_{1\to 2}(x_2)$$

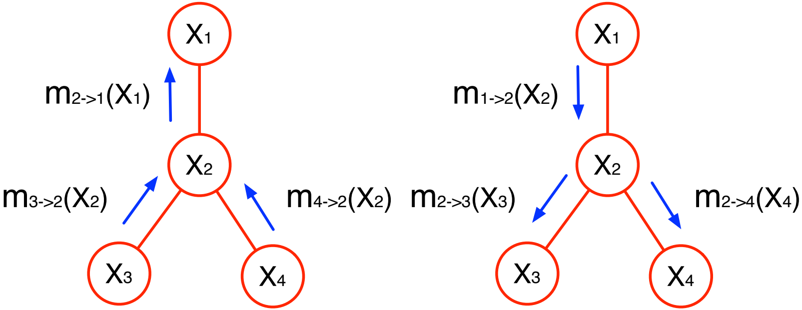
Inference - Message Passing

In general,

$$P(x_i) = \frac{1}{Z}\psi(x_i) \prod_{j \in Ne(i)} m_{j\to i}(x_i)$$

$$m_{j\to i}(x_i) = \sum_{x_j} \left(\psi(x_j)\psi(x_i, x_j) \prod_{k \in Ne(j)\setminus\{i\}} m_{k\to j}(x_j) \right)$$

Inference - sum-product



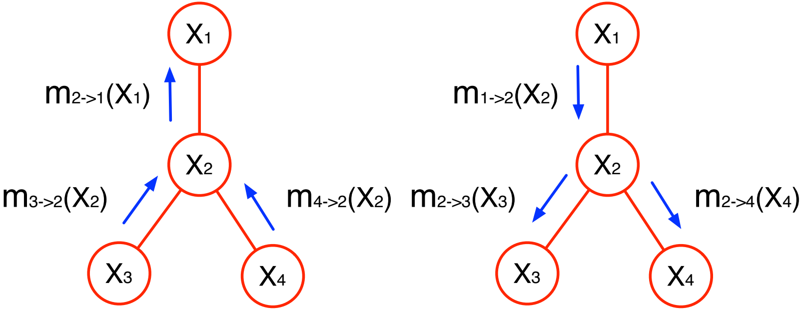
$$P(x_i) = \frac{1}{Z}\psi(x_i) \prod_{j \in Ne(i)} m_{j\to i}(x_i)$$

$$m_{j\to i}(x_i) = \sum_{x_j} \left(\psi(x_j)\psi(x_i, x_j) \prod_{k \in Ne(j)\setminus\{i\}} m_{k\to j}(x_j) \right)$$

called sum-product algorithm or belief propagation.

Inference - max-product

To compute $(x_1^*, \cdots, x_4^*) = \operatorname{argmax}_{x_1^*, \cdots, x_4^*} P(x_1^*, \cdots, x_4^*)$, use max-product algorithm.



$$x_i^* = \operatorname*{argmax}_{x_i} \left(\psi(x_i) \prod_{j \in Ne(i)} m_{j \to i}(x_i) \right)$$

$$m_{j \to i}(x_i) = \max_{x_j} \left(\psi(x_j) \psi(x_i, x_j) \prod_{k \in Ne(j) \setminus \{i\}} m_{k \to j}(x_j) \right)$$

Inference - Message Passing in Log Space

To avoid over/underflow,

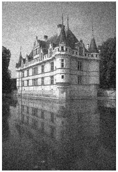
$$\log P(x_i) = \log(\psi(x_i)) + \sum_{j \in Ne(i)} \mu_{j \to i}(x_i) - \log(Z)$$

$$\mu_{j \to i}(x_i) := \log m_{j \to i}(x_i)$$

A real application

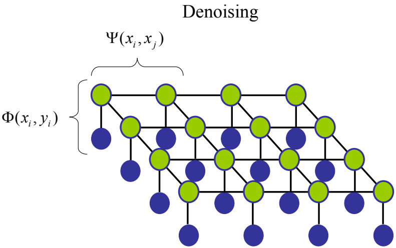
Denoising[1]

Original



Noisy

Corrected



$$X^* = \operatorname{argmax}_X P(X|Y)$$

[1] This example is from Tiberio Caetano's short course: "Machine Learning using Graphical Models"

More details of BP and other inference methods will be covered at the next talk.