Master in Computer Vision Barcelona

Module: Optimization methods in CV

Bayeasian networks and MRFs. Inference Problems. Message passing

Lecturer: O. Ramos Terrades

Goals of this Lecture & Tools

Goal

- Introduction to PGM: Bayesian networks and MRFs
- Inference Problems
- Message Passing

Tools

- UGM library at `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`
- OpenGM at `http://hci.iwr.uni-heidelberg.de/opengm2/`
  - Matlab and Python
  - Public benchmark at `http://hci.iwr.uni-heidelberg.de/opengm2/?l0=benchmark`

Outline

Introduction to PGM

Inference problems

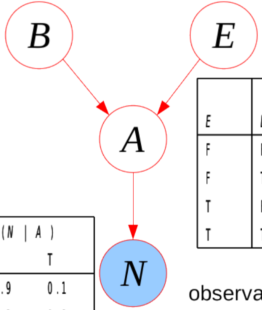
Inference algorithms

Message passing

- **Nodes**: (discrete) random variables
- **Edges**: represent conditional dependencies : $B \rightarrow A$

P(B) F	P(B) T
0.7	0.3

P(E) F	P(E) T
0.9	0.1



E	B	P(A | E,B) F	P(A | E,B) T
F	F	0.99	0.01
F	T	0.1	0.9
T	F	0.3	0.7
T	T	0.01	0.99

A	P(N | A) F	P(N | A) T
F	0.9	0.1
T	0.2	0.8

observation, evidence

B burglar
E earthquake
A alarm fires
N neighbor calls

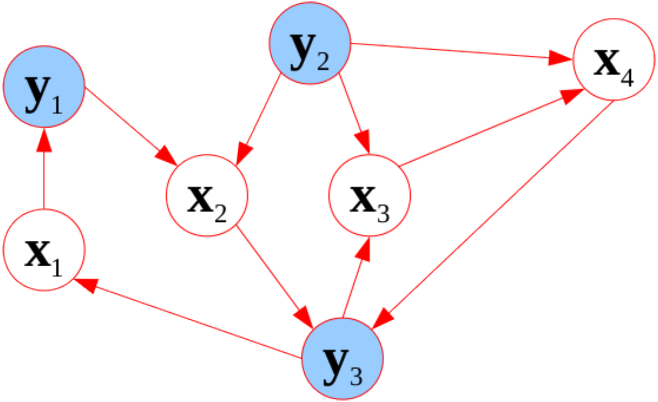
$$p(B|N = true)$$

Introduction to PGM: Avantages

- Tool for **modeling** complex systems in which uncertainty plays a role.
- Simple way to **visualize their structure** (dependencies).
- Inference can be expressed in terms of **algorithms** on a graph (mathematical expressions carried along implicitly) and computed efficiently
- Useful not only in **'pure' AI**: a number of computer vision and image processing problems can be posed as inference on a graphical model.

Outline

Inference problems: PGM types



- x_i **hidden** variable, unknown value: state, label ...
- y_i bf observed variables: $y_i = k$, $k \in \{0, \ldots L\}$

Inference problems (I)

Maximum *a Posteriori* (MAP)

- **Most probable state** of hidden variables $x_1, \ldots, x_N$, given observations $y_1, \ldots, y_M$

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x|y) \tag{1}$$

- **max-marginals**: most probable state for one, or more, hidden variable *independently*

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}}\, p(x_i|y) \tag{2}$$

Inference problems (II)

Marginal estimation

- $p(x_i)$ for one, or more, variables x_i. Also $p(x_i|y)$

$$p(x_i) = \int p(x_i, x_j)dx \quad \forall \text{ set of variables } i \text{ and } j. \tag{3}$$

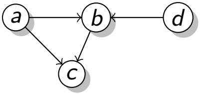
Marginal MAP

- **marginal-MAP** : most probable state for one, or more, hidden variable *independently*

$$\hat{x}_i = \underset{x_i}{\operatorname{argmax}} \int p(x_i, x_j)dx_j \tag{4}$$

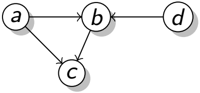
Inference problems: PGM types (II)

- Bayesian networks (BN): acyclic directed graphs, express **causality**.

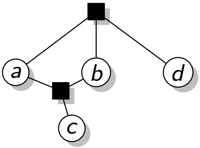


Inference problems: PGM types (II)

- Bayesian networks (BN): acyclic directed graphs, express **causality**.



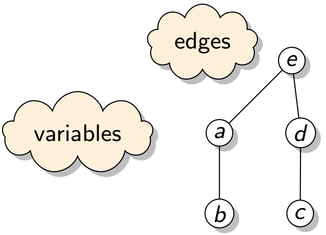
- Markov Random Fields (MRF): undirected graphs, express **compatibilities** between values of random variables.



Factor graphs

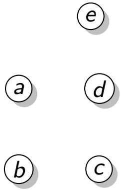
- In PGM with complex relations between random variables, we use **factor graphs**
- A PGM is factorized as a product of factor functions
- Factor graphs make this factorization more explicit through **factor nodes** (square nodes)

Graphs



Factors

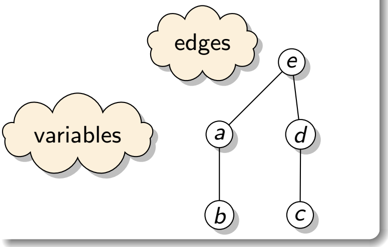
- Higher order relations



Factor graphs

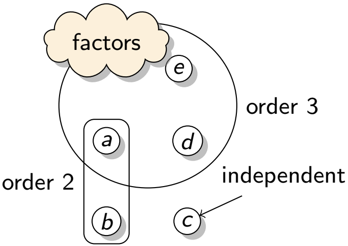
- In PGM with complex relations between random variables, we use **factor graphs**
- A PGM is factorized as a product of factor functions
- Factor graphs make this factorization more explicit through **factor nodes** (square nodes)

Graphs



Factors

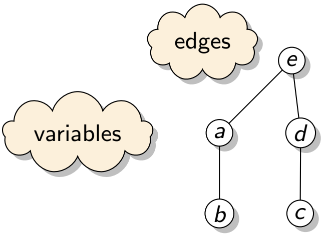
- Higher order relations



Factor graphs

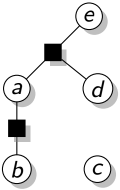
- In PGM with complex relations between random variables, we use **factor graphs**
- A PGM is factorized as a product of factor functions
- Factor graphs make this factorization more explicit through **factor nodes** (square nodes)

Graphs



Factors

- Higher order relations



Factorizes

$$p(a, b, c, d, e) = \frac{1}{Z}\phi(a, b)\phi(a, e, d)\phi(c) \tag{5}$$

Factor graphs

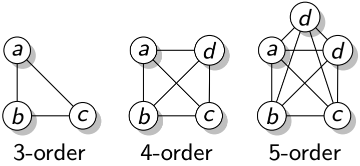
- Interactions defined on maximal cliques
- Factors defined on cliques
- Join pdfs factorizes on cliques:

$$p(x) = \frac{1}{Z} \prod_{\alpha} \phi_\alpha(x_\alpha) \quad (6)$$

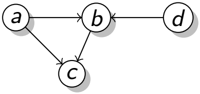
where $Z = \int p(x)dx$ is the partition function

Cliques

sub-graphs fully connected



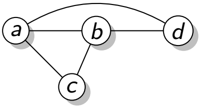
Bayes networks and factor graphs

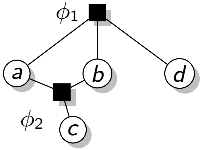


$$p(a,b,c,d) = p(c|a,b)p(a)p(b|a,d)p(d)$$

"Moralization"=marrying parents

- $p(c|a,b)$ a factor have to contain rvs a, b and c.
- $p(b|a,d)$ a factor have to contain rvs a, b and d.





$$p(a,b,c,d) = \frac{1}{Z}\phi(c,a,b)\phi(b,a,d) \qquad p(a,b,c,d) = \frac{1}{Z}\phi_1(c,a,b)\phi_2(b,a,d)$$

Outline

Introduction to PGM

Inference problems

Inference algorithms

Message passing

Inference algorithms: Message passing

Chain

$$x_0 \to x_1 \to x_2 \to x_3 \dashrightarrow x_{N-3} \to x_{N-2} \to x_{N-1} \to x_N$$

x_i, $i = 1, \dots, N$ discrete random variables (rv) of K possible states

Goal

Obtain the marginal $p(x_n)$ for a certain rv x_n. Observe that $p(x_n)$ is a vector in $\mathbb{R}^K$.

Naïve approach

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} \overbrace{p(x_1, \dots, x_N)}^{p(x)}$$

Inference algorithms: Message passing

Conditional independence: Markov property

$$x_{j-1} \rightarrow x_j \rightarrow x_{j+1}$$

$$x_{j-1} \perp x_{j+1} | x_j \Rightarrow p(x_j | x_{j+1}, x_{j-1}) = p(x_j | x_{j-1})$$

Factorization

$$\begin{aligned} p(x) &= p(x_N | x_{N-1}, \ldots, x_1) \cdots p(x_3 | x_2, x_1) p(x_2 | x_1) p(x_1) = \\ &= p(x_N | x_{N-1}) p(x_{N-1} | x_{N-2}) \cdots p(x_3 | x_2) p(x_2 | x_1) p(x_1) \end{aligned} \tag{7}$$

Problem

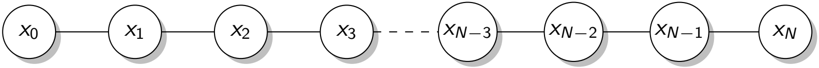
K^{N-1} terms $p(x)$, $N-1$ products each $\Rightarrow O(NK^N)$: exponential in the chain length

Inference algorithms: Message passing

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(x_N|x_{N-1})p(x_{N-1}|x_{N-2}) \cdots p(x_3|x_2)p(x_2|x_1)p(x_1)$$

factors: $\phi_{N-1,N}(x_{N-1}, x_N)$ $\phi_{N-2,N-1}(x_{N-2}, x_{N-1})$ $\phi_{2,3}(x_2, x_3)$ $\phi_{1,2}(x_1, x_2)$

joint distribution of an undirected graph with $Z = 1$.



Each time x_j varies, only two factors change:

$$\sum_{x_j} \boxed{}\boxed{x_j}\boxed{} = \boxed{}\boxed{}\sum_{x_j}\boxed{x_j}$$

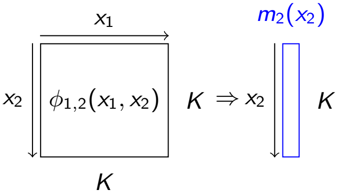
$\longrightarrow$ Distribute terms

$2K$ products 2 products

Inference algorithms: Message passing

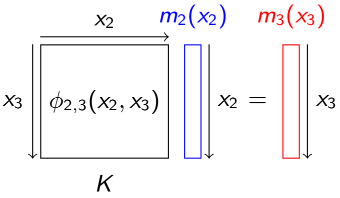
$$
\begin{aligned}
p(x_n) = \frac{1}{Z} & \left[\sum_{x_{n-1}} \phi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \phi_{2,3}(x_2, x_3) \sum_{x_1} \phi_{1,2}(x_1, x_2) \right] \right] \\
& \left[\sum_{x_{n+1}} \phi_{n,n+1}(x_{N-1}, x_N) \cdots \left[\sum_{x_{N-1}} \phi_{N-2,N-1}(x_{N-1}, x_{N-2}) \sum_{x_N} \phi_{N-1,N}(x_{N-1}, x_N) \right] \right]
\end{aligned} \tag{8}
$$

Sum columns



$m_2(x_2) = \sum_{x_1} \phi^t_{1,2}(x_1, x_2)$

Message passing

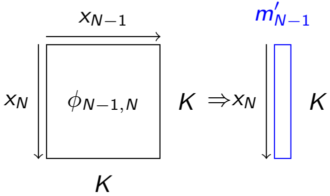


$m_3(x_3) = \phi_{2,3}(x_2, x_3) m_2(x_2)$

Inference algorithms: Message passing

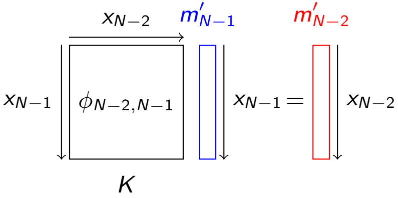
$$
\begin{aligned}
p(x_n) = \frac{1}{Z} & \left[\sum_{x_{n-1}} \phi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \phi_{2,3}(x_2, x_3) \sum_{x_1} \phi_{1,2}(x_1, x_2) \right] \right] \\
& \left[\sum_{x_{n+1}} \phi_{n,n+1}(x_{N-1}, x_N) \cdots \left[\sum_{x_{N-1}} \phi_{N-2,N-1}(x_{N-1}, x_{N-2}) \sum_{x_N} \phi_{N-1,N}(x_{N-1}, x_N) \right] \right]
\end{aligned} \tag{9}
$$

Sum columns



$m'_{N-1}(x_{N-1}) = \sum_{x_N} \phi_{N-1,N}$

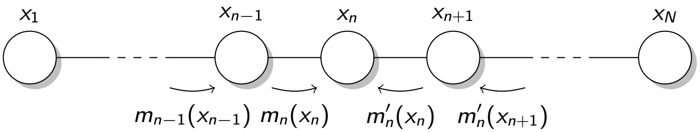
Message passing



$m'_{N-2} = \phi^t_{N-2,N-1} m'_{N-1}$

Inference algorithms: Message passing

- In total $N-2$ rvs and K^2 products each $\rightarrow O(NK^2)$: **linear with N** (before was $O(K^N)$) .
- This matrix computation is seen as a **message passing** process between rv on **undirected** graphs:

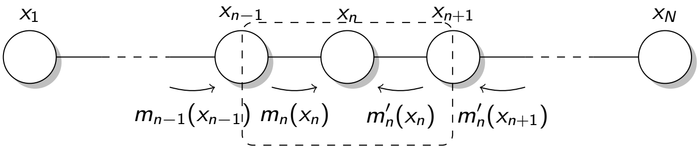


- to pass a message is a matrix product:

$$m_n(x_n) = \phi_{n-1,n}(x_{n-1}, x_n) m_{n-1}(x_{n-1}) \tag{10}$$

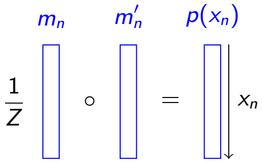
$$m'_n(x_n) = \phi^t_{n,n+1}(x_n, x_{n+1}) m'_{n+1}(x_{n+1}) \tag{11}$$

Inference algorithms: Message passing



- to estimate the marginal $p(x_n)$, x_n have to receive a message from left and right.

$$p(x_n) = \frac{1}{Z} m_n(x_n) m'_n(x_n) \Rightarrow$$

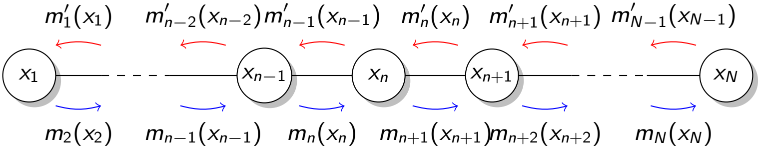


$\circ$: point-wise product

Inference algorithms: Message passing

To estimate $p(x_n)$ for all rvs:

- repeat N times the message passing: $O(N^2K^2)$
- share common messages: $O(NK^2)$



Partition function, Z:

$$Z = \prod_n Z_n \tag{12}$$

where $Z_n = \sum_{x_n} m_n(x_n) \circ m'_n(x_n)$