

Lecture 12 - scribbles

Tuesday, October 11, 2016
14:20

today: - UGM (cont)
- inference - elimination dg.

cond. indep. for UGM:

def: we say p satisfies **global Markov property** (with respect to undirected graph G)

iff $\forall A, B, S \subseteq V$ s.t. S separates A from B in G
disjoint then $X_A \perp\!\!\!\perp X_B \mid X_S$



S separates A & B
if all paths between A & B pass through S

prop: $p \in \mathcal{P}(G) \Rightarrow p$ satisfies the global Markov property

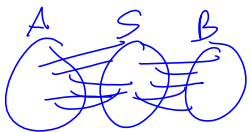
proof: WLOG, we assume $A \cap B \cap S = \emptyset$
and disjoint

(why? if not, let $\tilde{A} \triangleq A \cup \{a \in V : a \text{ and } A \text{ are not separated by } S\}$

$$\tilde{B} \triangleq V \setminus (S \cup \tilde{A})$$

[exercise: show that $\tilde{A} \perp\!\!\!\perp \tilde{B} \mid S \Rightarrow b$ separated from A]
(as we only know \tilde{A} & \tilde{B})

then if have $X_{\tilde{A}} \perp\!\!\!\perp X_{\tilde{B}} \mid X_S$; then $\Rightarrow X_A \perp\!\!\!\perp X_B \mid X_S$
by decomposition



let $C \in \mathcal{C}$; we can't have both that
 $C \cap A \neq \emptyset$ and $C \cap B \neq \emptyset$

$$\text{thus } p(x) = \frac{1}{Z} \prod_{\substack{C \in \mathcal{C} \\ C \cap A \neq \emptyset}} \psi_C(x_C) \prod_{\substack{C \in \mathcal{C} \\ C \cap B \neq \emptyset}} \psi_C(x_C)$$

$$= f(x_{A \cup S}) g(x_{B \cup S})$$

$$p(x_A \mid x_S) \propto p(x_{A \cup S})$$

$$\begin{aligned} &= \sum_{x_B} p(x_V) = \sum_{x_B} f(x_{A \cup S}) g(x_{B \cup S}) \\ &= f(x_{A \cup S}) \sum_{x_B} g(x_{B \cup S}) \end{aligned}$$

$$\Rightarrow p(x_A \mid x_S) = \underline{f(x_{A \cup S})}$$

$$\sum_{x'_A} f(x'_A, x_S)$$

similarly, $p(x_B | x_S) = g(x_{B|S}) / \sum_{x'_B} g(x'_B, x_S)$

$$p(x_A | x_S) p(x_B | x_S) = \frac{f(x_{A|S}) g(x_{B|S}) \{p(x_V)\}}{\sum_{x'_A} \sum_{x'_B} f(x'_A, x_S) g(x'_B, x_S) \{p(x_S)\}} = p(x_{A,B} | x_S)$$

ie. $X_A \perp\!\!\!\perp X_B \mid X_S$

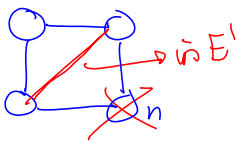
thm: (Hammersley-Chifford) if $p(x_V) > 0 \forall x_V$
 then $p \in \mathcal{L}(G) \iff p$ satisfies global Markov property
 (HC)

proof see ch. 16, use Möbius inversion formula
 (as exclusion-inclusion principle in prob.)

properties:

closure with respect to marginalization:

let $V' = V \setminus \{n\}$ $E' =$ edges in G + connect all neighbors of n in G together;



if $p \in \mathcal{L}(G)$; then $p(x_{1:n-1}) \in \mathcal{L}(G')$

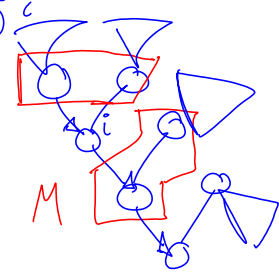
Directed GM vs undirected GM

def: Markov blanket for i is smallest set of nodes M
 (for graph G) s.t. $X_i \perp\!\!\!\perp X_V \mid X_M$


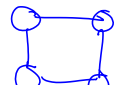

↑
rest

for UGM: $M = \{j : \{i, j\} \in E\}$ ie the neighbors of i

for DGM: $M = \pi_i \cup \text{children}(i) \cup \pi_j | j \in \text{children}(i)$



Recap: DGM | UGM

factorization:	$p(x) = \prod_i p(x_i x_{\pi_i})$	$\frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$
cond. indep.:	d-separation	separation
marginalization:	not closed in general () but fine for leaves	closed
cannot exactly capture:		 V-structure

Moralization

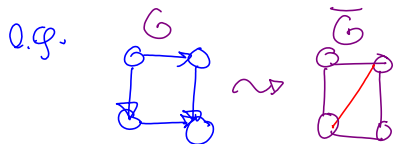
Let G be a DAG;
when can we transform to equivalent UGM?

def: for G a DAG, we call \bar{G} the moralized graph of G

where \bar{G} is an undirected graph with same V

and $\bar{E} = \{ \{i, j\} : (i, j) \in E \}$ } undirected version of E

$\cup \{ \{k, l\} : k \neq l \in \pi_i \text{ for some } i \}$ } "moralizations"
→ "marrying the parents"
← connect all the parents in big clique



↳ only add edges when $|\pi_i| \geq 2$ i.e. V-structure

prop: for DAG G with no v-structure [forest]

$$\text{then } \underset{\text{DGM}}{J(G)} = \underset{\text{UGM}}{J(\bar{G})}$$

but in general, can only say that $J(G) \subseteq J(\bar{G})$

[note that \bar{G} is the minimal undirected graph s.t. $J(G) \subseteq J(\bar{G})$]

Inference

say DGM $p(x) = \prod_i p(x_i | x_{\pi_i})$

UGM $p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$

moralization: $C = \{i\} \cup \pi_i$
 $\psi_C(x_C) \triangleq p(x_i | x_{\pi_i})$
 $Z = 1$

want to compute:

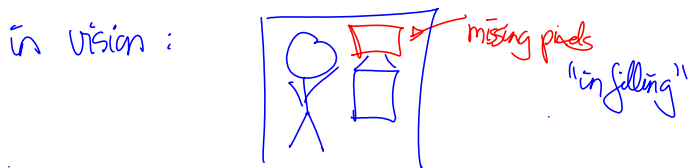
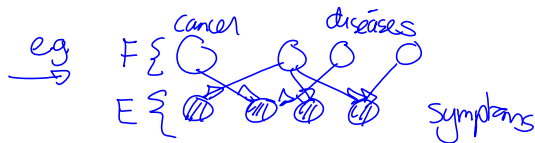
a) marginal: $p(x_F)$ for some FEV

b) conditional: $p(x_F | x_E)$
 "evidence"
 query (usually 1 or 2 nodes)

c) for UGM: partition function $Z = \sum_{x_V} \prod_{C \in \mathcal{C}} \psi_C(x_C)$

why? whole pt. of graph model is getting a model for $p(x_F | x_E)$
 query observation

- missing data
- prediction
- "latent cause"



also related inference: $\text{argmax}_{x_F} p(x_F | x_E)$
 could be big

inference is also needed during estimation (parameter fitting)
 (e.g. during E-step)

graph elimination alg.

• consider UGM $p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$

say want to compute $p(x_F)$ for F "query nodes"

$$x(a+b) = xa + xb$$

main trick: use distributivity

$$\sum_{x_1, x_2} f(x_1) g(x_2) = \left(\sum_{x_1} f(x_1) \right) \left(\sum_{x_2} g(x_2) \right)$$

can 'u yourself!

more generally: $\sum_{x_{1:n}} \prod_{i=1}^n f_i(x_i) = \prod_{i=1}^n \left(\sum_{x_i} f_i(x_i) \right)$

example:



$$\begin{aligned}
 p(x_4) &= \sum_{x_1, x_2, x_3} \frac{1}{Z} \Psi(x_1, x_2) \Psi(x_1, x_3) \Psi(x_2, x_3) \Psi(x_2, x_4) \\
 &= \frac{1}{Z} \sum_{x_3} \Psi(x_4, x_3) \underbrace{\sum_{x_2} \Psi(x_2, x_4) \sum_{x_1} \Psi(x_1, x_2) \Psi(x_1, x_3)}_{m_1(x_2, x_3)}
 \end{aligned}$$

"messages"

$m_1(x_2, x_3)$
factor with neighbors together

need to store as table

$$\begin{aligned}
 &\rightarrow m_2(x_3, x_4) \\
 &= \frac{1}{Z} m_2(x_4) \quad \text{last message is proportional to marginal } p(x_4) \\
 &\rightarrow \sum_{x_4} m_2(x_4) = Z
 \end{aligned}$$

general alg: graph Eliminate

- init
- a) choose an elimination ordering s.t. F are the last nodes
 - b) put all $\Psi_c(x_c)$ on "active list"

- "update"
- c) repeat in order of variables to eliminate
 - 1) remove all factors in active list with x_i in it & take the product

$$i.e. \prod_{\alpha: x_i \in \alpha} \Psi_\alpha(x_\alpha)$$

- 2) sum x_i to get new factor

i.e. let S_i be variables of these factor excluding x_i

$$\text{get } m_i(x_{S_i}) \triangleq \sum_{x_i} \prod_{\alpha} \Psi_\alpha(x_\alpha) \quad \begin{matrix} \text{new clique to} \\ \text{sum} \\ \text{over } S_i \cup \{x_i\} \end{matrix}$$

$|S_i| \triangleq |\cup \alpha \setminus \{x_i\}|$

- 3) put back $m_i(x_{S_i})$ in active list

new table to store

- "normalize"
- d) last factor left has only $x_F \Rightarrow$ proportional to $p(x_F)$

question: memory needed? $\approx 2^{|S_i|} \cdot \# \text{ factors}$

computational cost $\approx O(2^{(s+1)})$