

Lecture 13 - scribbles

Tuesday, October 16, 2018 14:20

today: • finish UGM
• inference

general themes in this class

A) representation $\begin{matrix} \rightarrow \text{DGM} \\ \rightarrow \text{UGM} \end{matrix}$

parametrization \rightarrow exponential family

} modeling high dim. distribution

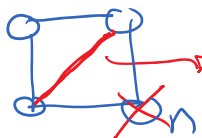
B) inference computing $p(x_Q | x_E)$
"query" "evidence"

\rightarrow today: elimination alg.
 \rightarrow sum-product / belief propagation

C) statistical estimation \rightarrow MLE
maximum entropy
method of moments

back to UGM

properties: \rightarrow closure with respect to marginalization



let $V' = V \setminus \{n\}$ $E' =$ edges in $G \setminus \{n\}$

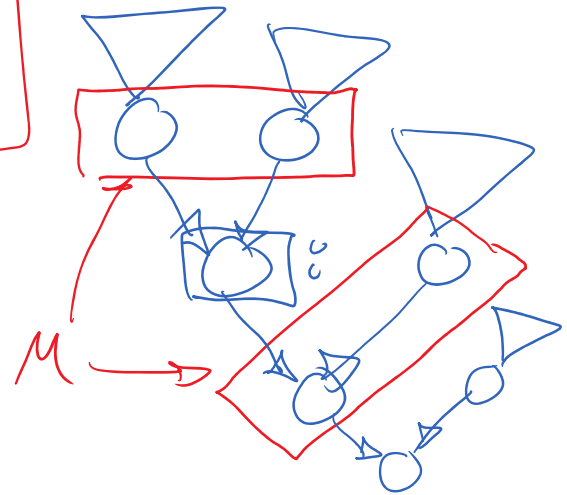
+ connect all neighbors of n in G together (new edge)




$$\{\text{marginal on } x_{1:n-1} \text{ for } p \in \mathcal{S}(G)\} = \mathcal{S}(G')$$

DGM vs. UGM

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

- for UGM: $M = \{j : \{i, j\} \in E\}$ = set of neighbors of i
- for DGM: $M = \pi_i \cup \text{children}(i) \cup \bigcup_{j \in \text{children}(i)} \pi_j$



recap:	DGM	UGM
factorization:	$p(x) = \prod_i p(x_i x_{\pi_i})$	$\frac{1}{Z} \prod_C \psi_C(x_C)$
cond. indep.:	d-separation	separation
marginalization:	not closed in general e.g.  but is fine for leaves	closed
cannot exactly capture some:	 families	 v-structure

Moralization

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

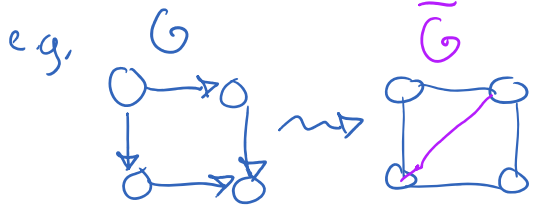
def: for G a DAG, we call \bar{G} the moralized graph of G

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

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

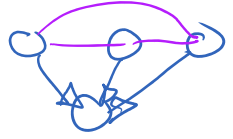
$\cup \{ \{k,l\} : k \neq l \in \pi_i \text{ for some } i \}$ } "moralization"

"marrying the parents" (⚭)



↑ connect all the parents of i with i in big cliques

↳ only need to add edges when $|\pi_i| > 1$
i.e. v-structure



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

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

but in general, can only say that $\underset{\text{DAG}}{I(G)} \subseteq \underset{\text{UGM}}{I(\bar{G})}$

[note that \bar{G} is the minimal undirected graph s.t. $\underset{\text{DAG}}{I(G)} \subseteq \underset{\text{UGM}}{I(\bar{G})}$]

Inference:

present alg. for UGM

[for simplicity and more generality]

but note that sometimes more efficient to work with DGM directly

make DGM \rightsquigarrow \subseteq UGM via moralization

ie. DGM: $p(x) = \prod_i p(x_i | \pi_i)$

moralize: $C = \{c\} \cup \pi_c$

$= \prod_{z=1}^K \prod_{c \in C} \psi_c(x_c)$

$\psi_c(x_c) \triangleq p(x_i | \pi_i)$

inference: want to compute:

a) marginal: $p(x_F)$ for some $F \subseteq V$

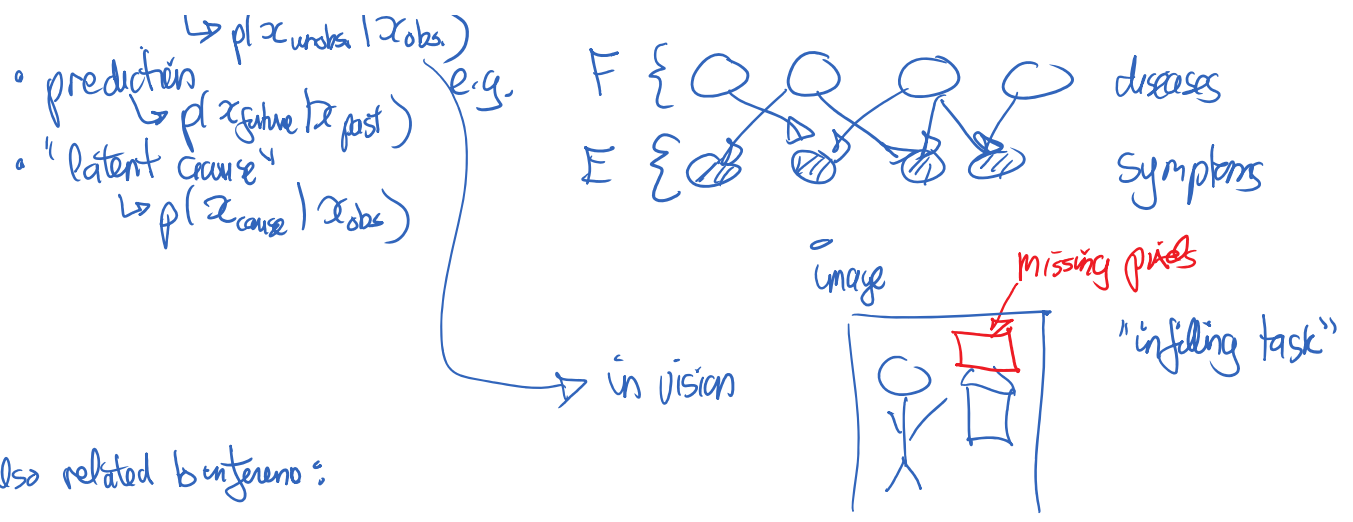
b) conditional: $p(x_F | x_E)$
 "query" "evidence"

c) for UGM: partition function $Z = \sum_{x \in \mathcal{X}} \prod_{c \in C} \psi_c(x_c) \quad C \subseteq V$

why?

- missing data $\hookrightarrow p(x_{unobs} | x_{obs})$
- prediction e.g.

$F \{ \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \text{ diseases} \}$



* also related benchmarks:

$$\operatorname{argmax}_{x_F} p(x_F | x_E)$$

x_F could be big here (e.g. speech recognition)

* inference is also needed during estimation (parameter fitting MLE)
(oc during E-step $p(z|x)$)

15h31

graph elimination algo (for inference)

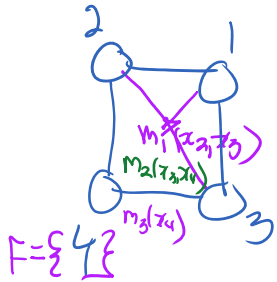
- consider $p \in \mathcal{S}(G)$ \downarrow undir. $p(x) = \frac{1}{Z} \prod_{c \in G} \psi_c(x_c)$

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

main trick: use distributivity of \oplus over $\odot \rightarrow C \odot (a \oplus b) = C \odot a + C \odot b$

$$\sum_{x_1, x_2} f(x_1) g(x_2) = \left(\sum_{x_1} f(x_1) \right) \left(\sum_{x_2} g(x_2) \right) \quad [\text{convince yourself?}]$$

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



$$p(x_4) = \frac{1}{Z} \sum_{x_1, x_2, x_3} \psi(x_1, x_2) \psi(x_2, x_3) \psi(x_3, x_4) \psi(x_2, x_4)$$

$$= \frac{1}{Z} \sum_{x_3} \psi(x_4, x_3) \sum_{x_2} \psi(x_2, x_4) \underbrace{\sum_{x_1} \psi(x_1, x_2) \psi(x_1, x_3)}_{m_1(x_2, x_3) \text{ "message"}}$$

$$\underbrace{\sum_{x_2} \psi(x_2, x_4) m_1(x_2, x_3)}_{m_2(x_3, x_4) \text{ (stored as a table)}}$$

$$= \frac{1}{Z} m_3(x_4)$$

last message is proportional to marginal $p(x_4)$

$$\sum_{x_4} m_3(x_4) = Z$$

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 (say x_i is variable to eliminate)

"update"

c) repeat in order of variables to eliminate
(say x_i is variable to eliminate)

1) remove all factors from active list with x_i in π \rightarrow take product

$$\text{i.e. } \prod_{\alpha \text{ s.t. } i \in \alpha} \gamma_{\alpha}(x_{\alpha})$$

2) sum x_i to get a new factor $m_i(x_{S_i})$

i.e. S_i all variables in those factors except i

$$\text{get } m_i(x_{S_i}) \triangleq \sum_{x_i} \prod_{\alpha} \gamma_{\alpha}(x_{\alpha})$$

$$S_i = (\cup_{\alpha \text{ s.t. } i \in \alpha} \alpha) \setminus \{i\}$$

new clique to sum over $S_i \cup \{i\}$

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 $p(x_F)$

memory needed? $\approx 2^{\max_i |S_i|} \cdot (\# \text{ factors})$

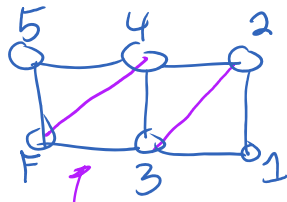
computational cost $\approx 2^{\max_i |S_i| + 1} \cdot n$

graph of cliques...

cliques formed during graph elimination



(*) running graphElim + keep track of all edges added

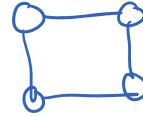


'augmented graph' after graphEliminate

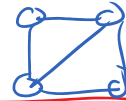
* Running graphElim + keep track of all edges added

yields a triangulated graph

def: graph with no cycle of size 4 or more that cannot be broken by a "chord"



↪ not triangulated



↪ triangulated

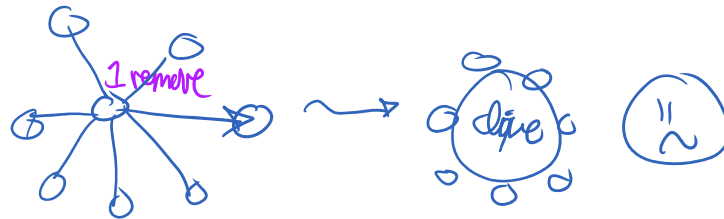
treewidth of a graph $\triangleq \min_{\text{over all elimination orderings}} \{ \text{size of biggest clique formed} - 1 \}$

↳ convention treewidth (tree) = 1

both memory & running time of graphEliminate is dominated by $2^{\text{size of biggest clique} + 1}$

best ordering guess $\approx 2^{\text{treewidth} + 2}$

not all orderings are good



bad news:

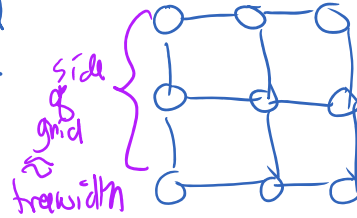
a) NP hard to compute treewidth (or find best ordering)

b) NP hard to do (exact) inference in general OGM

⇒ need approx. methods

example: treewidth of a grid

$$\approx \sqrt{|V|}$$



good news:

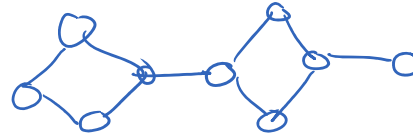
* inference in linear time for trees
(HMM, Markov chain)

* efficient for "small treewidth graph"
use
→ junction tree alg.

$$\rightarrow |V| + |E|$$

(treewidth = 1)

("sum-product alg.")



16/25