Lecture 14 - Inference in UGM

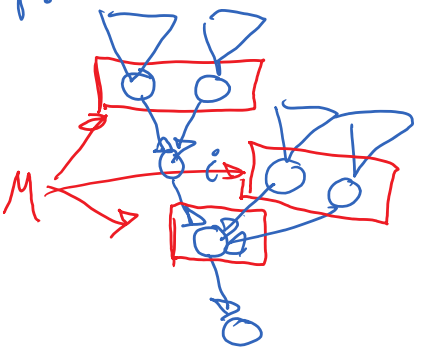
October 26, 2021 4:20 PM

today : · DGM vs. UGM
· inference

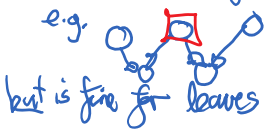
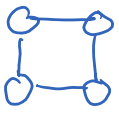
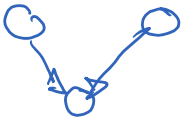
DGM vs. UGM

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

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



recap:

	DGM	UGM
factorization:	$p(x) = \prod_i p(x_i \mid x_{\pi_i})$	$\frac{1}{Z} \prod_C \psi_C(x_C)$
cond. indep.:	d-separation	separation
marginalization:	not closed in general (e.g.  but is fine for leaves)	closed (connect all neighbors of removed node)
cannot exactly capture same families		

Moralization:

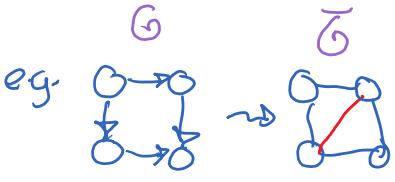
Let G be a DAG; when can we get an equivalent UGM?

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

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

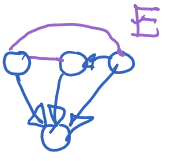
and $\overline{E} = \{\{i,j\} : (i,j) \in E\}$ } undirected version of E

$\cup \{\{k,l\} : k \neq l \in \pi_i$ for some $i\}$ } "moralization" — "marrying the parents" (:D)

e.g. 

↳ connect all the parents of i with i in big clique

↳ only needed if $|\pi_i| > 1$



any node ... ie. v-structure

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

then $\mathcal{I}(G) = \mathcal{I}(\bar{G})$ (DGM, UGM)

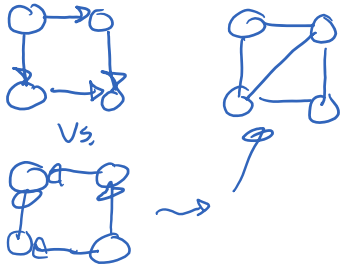
(proof: see assignment?)

but in general, can only say $\mathcal{I}(G) \subseteq \mathcal{I}(\bar{G})$ (DAG)

[note that $\bar{G}$ is the minimal undirected graph G' s.t. $\mathcal{I}(G) \subseteq \mathcal{I}(G')$]

$$p(x) = \prod_i p(x_i \mid x_{\pi_i}) \quad \text{over } \bar{G}$$

$p(x_i \mid x_{\pi_i}) \to \psi_C(x_C)$ where $C = \{i\} \cup \pi_i$



General themes in this class

A) representation → DGM / UGM

parameterization → exponential family

} model high dim. distribution

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

→ today: elimination algorithm

→ next: sum-product / belief propagation (for trees)

later: approximate inference e.g. MCMC variational

C) statistical estimation / learning → MLE, maximum entropy, method of moments

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 $Z = \sum_{x_V} \left(\prod_C \psi_C(x_C) \right)$
exponential sum

why? • missing data : $p(x_{unobs} | x_{obs})$
↳ example

• prediction $p(x_{future} | x_{past})$
• "latent cause" $p(x_{cause} | x_{obs})$

* also related inference

$\text{argmax}_{x_F} \; p(x_F | x_E)$
F could be big (Viterbi alg.) (e.g. speech recognition)

* inference is also needed during estimation (parameter fitting MLE)
[e.g. during E-step $p(z|x)$]

* present inference alg. for UGM [for simplicity and more generality
but note that sometimes it's more efficient to
work directly with DGM]

make DGM → ⊆ UGM via moralization

ie. $p \in DGM$; $p(x) = \prod_i p(x_i | x_{\pi_i})$ moralize : $C_i \triangleq \{i\} \cup \pi_i$

$= \frac{1}{Z} \prod_i \psi_{C_i}(x_{C_i})$ $\psi_{C_i}(x_{C_i}) \triangleq p(x_i | x_{\pi_i})$

$Z = 1$

$p(x_1) = \sum_{x_{2:n}} p(x) = \sum_{x_{2:n}} \prod_i \frac{1}{Z} \psi_{C_i}(x_{C_i})$

17h37

graph elimination alg (for inference in generic UGM)

• consider $p \in \mathcal{G}(\mathcal{G})$ $p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$
undirected
say want to compute $p(x_F)$ for $F \subseteq V$ "query nodes"

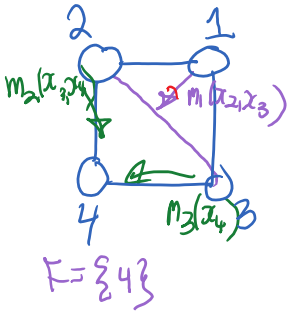
main trick : use distributivity of $\oplus$ over $\odot$ ⇝ $c \odot (a \oplus b) = c \odot a \oplus c \odot b$

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

more generally

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

$O(k^n \cdot n)$ $\quad$ $O(k \cdot n)$



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

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

$\sum_{x_1}\psi(x_1,x_2)\psi(x_1,x_3) = m_1(x_2,x_3)$ ("message")

stored as table

$\sum_{x_2}\psi(x_2,x_4)\,m_1(x_2,x_3) = m_2(x_3,x_4)$

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

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

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

$$\boxed{p(x_4) = \frac{m_3(x_4)}{Z}}$$

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)

1) remove all factors from active active list with x_i in it & take their product

ie. $\prod_{\alpha \text{ s.t. } i \in \alpha} \psi_\alpha(x_\alpha)$

2) sum over x_i to get a new factor $m_i(x_{S_i})$ (think as $\psi_{S_i}(x_{S_i})$)

ie. S_i are all variables in these factors except i

get $m_i(x_{S_i}) \triangleq \sum_{x_i} \prod_{\alpha \text{ s.t.}} \psi_\alpha(x_\alpha)$

get $m_i(x_{S_i}) \triangleq \sum_{x_i} \prod_{\alpha \text{ s.t. } i \in \alpha} \psi_\alpha(x_\alpha)$

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

$S_i \triangleq \left(\bigcup_{\alpha \text{ s.t. } i \in \alpha} \alpha\right) \setminus \{i\}$

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

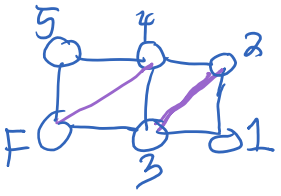
"normalize" d) last product of factors left has only $x_F \Rightarrow$ proportional $p(x_F)$

suppose $x_i \in \{x_1, \dots\}$ memory needed? $\approx 2^{\max_i |S_i|} \cdot (\#\text{ of factors})$

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

later, related "treewidth" of a graph

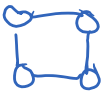
(*) "augmented graph" → graph obtained by running graph eliminate + keeping track of all edges added (for a fixed ordering)



note: augmented graph after graph Eliminate is always a triangulated graph

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

an edge between two non-neighboring nodes in cycle


← not triangulated


△

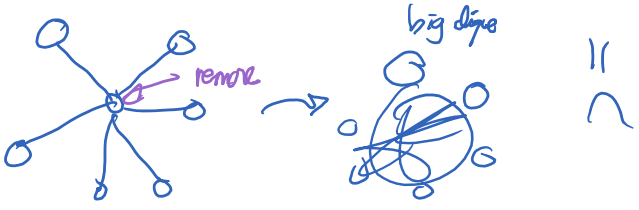
treewidth of a graph $\triangleq$ min over all possible elimination orderings { size of biggest clique in augmented graph -1 }

convention treewidth(tree) = 1

both memory & running time of graph Eliminate is dominated by $2^{\text{size of biggest clique}}$

best ordering gives $\approx 2^{\text{treewidth}+1}$

not all orderings are good



bad news:

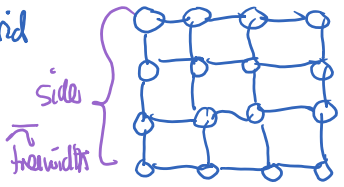
i) NP-hard to compute treewidth of general graph (or find best ordering)

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

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

$\Rightarrow$ approximate methods

example:
treewidth of a grid
$\approx \sqrt{|V|}$



good news

* inference is linear time for trees (treewidth = 1) ("sum-product alg.")
  → $|V| + |E|$ (HMM, Markov chain, etc.)
* efficient for "small treewidth graph"
  → use junction tree alg.