Readings:
K&F: 9.1, 9.2, 9.3, 9.4, 4.1, 4.2, 4.3, 4.4

Junction Trees 3

Undirected Graphical Models

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University
October 27th, 2008

Introducing message passing with division

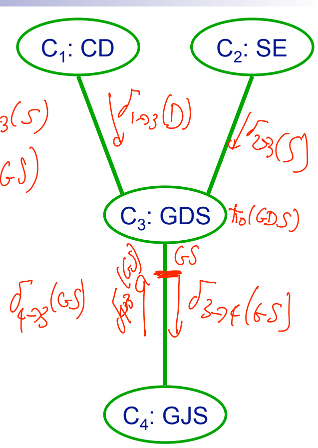
- Variable elimination (message passing with multiplication)
  - message: $\delta_{3\to4}(GS) = \sum_d \Pi_0(Gds)\,\delta_{1\to3}(d)\,\delta_{2\to3}(S)$
  - at convergence
  - belief: $P(GDS) = \Pi_0(GDS)\,\delta_{1\to3}(D)\,\delta_{2\to3}(S)\,\delta_{4\to3}(GS)$

node 3

- Message passing with division:
  - Belief: $\Pi_0(GDS)\,\delta_{1\to3}\,\delta_{2\to3}\,\delta_{4\to3}$
  - Belief about separator: $\sum_d \Pi_0(Gds)\,\delta_{1\to3}(d)\,\delta_{2\to3}(S)\,\delta_{4\to3}(GS)$

    remove content from 4
  - message:

    Sum $\underbrace{\sum_d \Pi_0(Gds)\,\delta_{1\to3}\,\delta_{2\to3}\,\cancel{\delta_{4\to3}}}_{\cancel{\delta_{4\to3}}}$



Factor division

- Let **X** and **Y** be disjoint set of variables
- Consider two factors: $\phi_1(\mathbf{X},\mathbf{Y})$ and $\phi_2(\mathbf{Y})$
- Factor $\psi=\phi_1/\phi_2$
  - 0/0=0

a^1	b^1	0.5
a^1	b^2	0.2
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.45

a^1	0.8
a^2	0
a^3	0.6

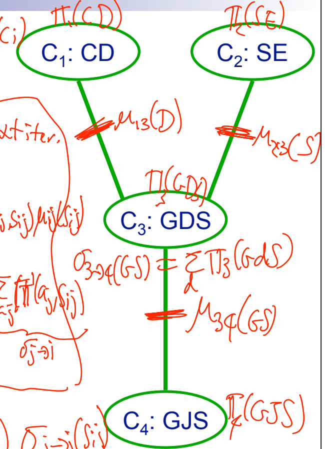
a^1	b^1	0.625
a^1	b^2	0.25
a^2	b^1	0
a^2	b^2	0
a^3	b^1	0.5
a^3	b^2	0.75

$$\psi^{(X,Y)} = \frac{\phi_1(X,Y)}{\phi_2(Y)}$$

$$\psi(X^{t}, Y^{f}) = \frac{\phi_1(X=t, Y=f)}{\phi_2(Y=f)}$$

Lauritzen-Spiegelhalter Algorithm (a.k.a. belief propagation)

Simplified description see reading for details



$\pi_i(C_i) \overset{init}{=} \pi_0(C_i)$

- Separator potentials μ_{ij}
  - one per edge (same both directions)
  - holds "last message"
  - initialized to 1

$\mu_{ij} \overset{init}{=} 1$

$C_i \; \overset{S_{ij}}{—} \; C_j$ $\quad C_i / S_{ij} = A_i$

(*) next iter, $j \to i$: $\sum_{a_j} \pi_j'(a_j, s_{ij}) \mu_{ij}(s_{ij})$ $\quad \underbrace{\mu_{ij}(s_{ij})}_{\sigma_{i\to j}} \underbrace{\sum_{c_j} \pi_i'(a_j, s_{ij})}_{\sigma_{j \to i}}$

- Message ~~i→j~~ $i \to j$
  - what does i think the separator potential should be? wrt j
    - $\sigma_{i\to j}(s_{ij}) = \sum_{a_i} \pi_i(a_i, s_{ij})$
  - update belief for j:
    - pushing j to what i thinks about separator

$\pi_j'(C_j) \leftarrow \pi_j(C_j) \frac{\sigma_{i\to j}(s_{ij})}{\mu_{ij}(s_{ij})}$

  - replace separator potential:

$\mu_{ij}'(s_{ij}) \leftarrow \sigma_{i\to j}(s_{ij})$ (*)

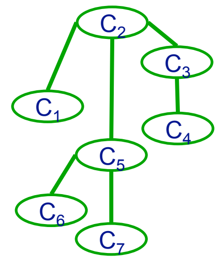
$\pi_1(CD)$, $\pi_2(SE)$, $\mu_{13}(D)$, $\mu_{23}(S)$, $\pi_3(GDS)$, $\sigma_{3\to4}(GS) = \sum_{d} \pi_3(GdS)$, $\mu_{34}(GS)$, $\pi_4(GJS)$

Convergence of Lauritzen-Spiegelhalter Algorithm

- Complexity: Linear in # cliques
  - for the "right" schedule over edges (leaves to root, then root to leaves)
- **Corollary**: At convergence, every clique has correct belief

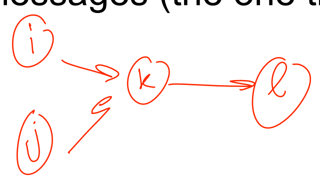
$$\pi_i(C_i) = P(C_i)$$

$$\mu_{ij}(S_{ij}) = P(S_{ij})$$



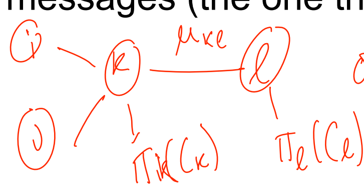
VE versus BP in clique trees

- VE messages (the one that multiplies) $C_k = A_k \cup S_{kl}$



$$\delta_{k\to l} = \sum_{a_k} \pi_0(C_k)\, \delta_{i\to k}\, \delta_{j\to k}$$

- BP messages (the one that divides) $k \to l$



$$\delta_{k\to l} = \sum_{a_k} \pi_k(a_k, S_{kl})$$

$$\pi_l'(C_l) \leftarrow \frac{\pi_l(C_l)\, \delta_{k\to l}}{\mu_{kl}}$$

Clique tree invariant

- **Clique tree potential**:
  - Product of clique potentials divided by separators potentials

$$\pi_T(X) = \frac{\prod_i \pi_i(C_i)}{\prod_{ij} \mu_{ij}(S_{ij})}$$

$\pi_0(C_i) \equiv$ product of CPTs assigned to node i

- Clique tree **invariant**:
  - $P(\mathbf{X}) = \pi_T(\mathbf{X})$

at initialization: $\mu_{ij} = 1$, $\pi_i(C_i) = \pi_0(C_i)$

$$\pi_T(X) = \prod_i \pi_0(C_i) = \prod_i P(X_i \mid Pa_{X_i}) = P(X)$$

Belief propagation and clique tree invariant

$AB \leftarrow P(AB)$ — $BC \leftarrow P(BC)$ — $CD \leftarrow P(CD)$; separators $P(B)$, $P(C)$

- **Theorem**: Invariant is maintained by BP algorithm!

at iter. t, I have $\pi_i^{(t)}$, $\mu_{ij}^{(t)}$

sent message $i \to j$ at $t+1$:

$$\pi_j^{(t+1)} \leftarrow \pi_j^{(t)} \frac{\sigma_{i\to j}^{(t)}}{\mu_{ij}^{(t)}}$$

$$\mu_{ij}^{(t+1)} \leftarrow \sigma_{ij}^{(t)}$$

$$\frac{\pi_T^{(t+1)}(X)}{\pi_T^{(t)}(X)} = \frac{\pi_j^{(t+1)}}{\mu_{ij}^{(t+1)}} \cdot \frac{\mu_{ij}^{(t)}}{\pi_j^{(t)}} = 1$$

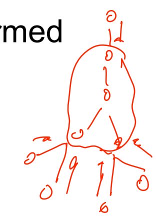
- BP reparameterizes clique potentials and separator potentials
  - At convergence, potentials and messages are marginal distributions $\pi_i(C_i) = P(C_i)$ $\mu_{ij}(S_{ij}) = P(S_{ij})$

$$P(X) = \pi_T(X) = \frac{\prod_i P(C_i)}{\prod_{ij} P(S_{ij})}$$

fundamental eqn. of junction trees

Subtree correctness

- **Informed message** from i to j, if all messages into i (other than from j) are informed
  - Recursive definition (leaves always send informed messages)
- **Informed subtree**:
  - All incoming messages informed
- **Theorem**:
  - Potential of connected informed subtree *T'* is marginal over scope[*T'*]
- **Corollary**:
  - At convergence, clique tree is *calibrated*
    - π_i = P(scope[π_i])
    - μ_{ij} = P(scope[μ_{ij}])



building

Clique trees versus VE for inference

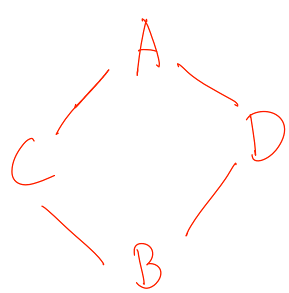
- Clique tree advantages
  - Multi-query settings
  - Incremental updates
  - Pre-computation makes complexity explicit
- Clique tree disadvantages
  - Space requirements – no factors are “deleted”
  - Slower for single query
  - Local structure in factors may be lost when they are multiplied together into initial clique potential

Clique tree summary

- Solve marginal queries for all variables in only twice the cost of query for one variable
- Cliques correspond to maximal cliques in induced graph
- Two message passing approaches
  - VE (the one that multiplies messages)
  - BP (the one that divides by old message)
- Clique tree invariant
  - Clique tree potential is always the same
  - We are only reparameterizing clique potentials
- Constructing clique tree for a BN
  - from elimination order
  - from triangulated (chordal) graph
- Running time (only) exponential in size of largest clique
  - Solve **exactly** problems with thousands (or millions, or more) of variables, and cliques with tens of nodes (or less)

Swinging Couples revisited

- This is no perfect map in BNs
- But, an undirected model will be a perfect map

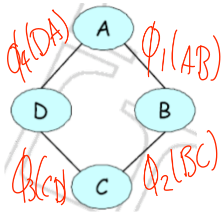


$C \perp D \mid AB$

$A \perp B \mid CD$

Potentials (or Factors) in Swinging Couples

$t=1 \quad f=0$



$\phi_1[A,B]$			$\phi_2[B,C]$			$\phi_3[C,D]$			$\phi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

≈ 200

$P(ABCD) \propto \phi_1(AB)\,\phi_2(BC)\,\phi_3(CD)\,\phi_4(DA)$

every clique (max) is associated with a potential

$P(ABCD) = \frac{1}{Z}\,\phi_1\,\phi_2\,\phi_3\,\phi_4$

Z normalization, partition function

$P(A=t, B=f, C=t, D=f) \propto 1 \times 1 \times 100 \times 1$ } same

$P(A=f, B=f, C=t, D=t) \propto 1 \times 1 \times 1 \times 100$

$1 \times 1 \times 1 \times 200$ twice as likely

Computing probabilities in Markov networks v. BNs

- In a BN, can compute prob. of an instantiation by multiplying CPTs

$P(X) = \prod_i P(X_i \mid Pa_{X_i})$

- In an Markov networks, can only compute ratio of probabilities directly

$\phi_1[A,B]$			$\phi_2[B,C]$			$\phi_3[C,D]$			$\phi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Normalization for computing probabilities

- To compute actual probabilities, must compute normalization constant (also called partition function)

$$P(ABCD) = \frac{1}{Z}\,\phi_1(AB)\,\phi_2(BC)\,\phi_3(CD)\,\phi_4(DA)$$

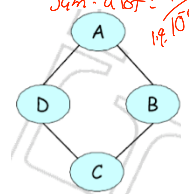
$$Z = \sum_a \sum_b \sum_c \sum_d \phi_1(a,b)\,\phi_2(b,c)\,\phi_3(c,d)\,\phi_4(d,a)$$

Assignment				Unnormalized (Potential)	Normalized
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5000000 a lot	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014

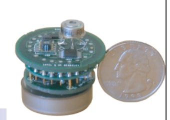
Sum = a lot = $\frac{10}{1.4 \cdot 10^{-6}}$

- Computing partition function is hard! → Must sum over all possible assignments

Can use VE to compute Z
if Markov Network has low tree width

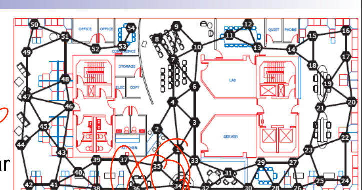


Factorization in Markov networks



- Given an undirected graph H over variables $\mathbf{X}=\{X_1,\ldots,X_n\}$
- A distribution P **factorizes** over H if ~~∄~~ ∃ (D_i, D_j may overlap)
  - subsets of variables $\mathbf{D}_1 \subseteq \mathbf{X}, \ldots, \mathbf{D}_m \subseteq \mathbf{X}$, such that the $\mathbf{D}_i$ are *fully connected* in H
  - *non-negative potentials* (or factors) $\phi_1(\mathbf{D}_1), \ldots, \phi_m(\mathbf{D}_m)$
    - also known as clique potentials
  - such that

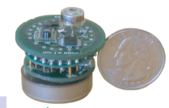
$$P(X) = \frac{1}{Z}\prod_{i=1}^{m} \phi_i(D_i)$$



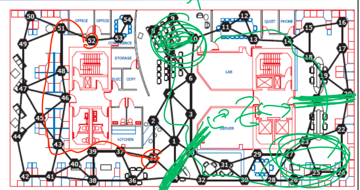
$D_1 = \{1, 34, 35\}$
$D_2 = \{36, 37\}$
$D_3 = \{34, 35, 36\}$

- Also called Markov random field H, or Gibbs distribution over H

Global Markov assumption in Markov networks



- A path $X_1 - \ldots - X_k$ is **active** when set of variables **Z** are observed if none of $X_i \in \{X_1,\ldots,X_k\}$ are observed (are part of **Z**)



X

Y

$sep_H(X;Y|Z)$

- Variables **X** are **separated** from **Y** given **Z** in graph *H*, $sep_H(\mathbf{X};\mathbf{Y}|\mathbf{Z})$, if there is no active path between any $X \in \mathbf{X}$ and any $Y \in \mathbf{Y}$ given **Z**

- The **global Markov assumption** for a Markov network *H* is

$$sep_H(X;Y|Z) \Rightarrow X \perp Y | Z$$