Graphical Models

Clique trees & Belief Propagation

Siamak Ravanbakhsh

Fall 2019
Learning objectives

- message passing on clique trees
- its relation to variable elimination
- two different forms of belief propagation
Recap: variable elimination (VE)

• marginalize over a subset - e.g., \( P(J) = \sum_{C,D,I,G,S,L,H} P(C, D, I, G, S, L, J, H) \)
• expensive to calculate (why?)
• use the factorized form of \( P \)

\[
\sum_{C,D,I,G,S,L,H} P(D|C)P(G|D,I)P(S|I)P(L|G)P(J|L,S)P(H|G,J)
\]
Recap: variable elimination (VE)

- marginalize over a subset - e.g., \( P(J) = \sum_{C,D,I,G,S,L,H} P(J,H,C,D,I,G,S,L) \)
- expensive to calculate (why?)
- use the factorized form of \( P \)

\[
\sum_{C,D,I,G,S,L,H} P(D|C)P(G|D,I)P(S|I)P(L|G)P(J|L,S)P(H|G,J)
\]

\( \phi_2(H,G,J) \)

think of this as a factor/potential

same treatment of

- Bayes-nets
- Markov nets

for inference

note that they do not encode the same CIs
Recap: variable elimination (VE)

- marginalize over a subset - e.g., \( P(J) = \sum_{C, D, I, G, S, L, H} P(J, H, C, D, I, G, S, L) \)
- expensive to calculate (why?)
- use the factorized form of \( P \)

\[
\sum_{C, D, I, G, S, L} \phi_1(D, C) \phi_2(G, D, I) \phi_3(S, I) \phi_4(L, G) \phi_5(J, L, S) \phi_6(H, G, J)
\]

\[
= \ldots \sum_I \phi_3(S|I) \sum_D \phi_2(G, D, I) \sum_C \phi_1(D, C)
\]

- repeat this

\[
\psi_1(D) \quad \psi_1(D, C')
\]

\[
= \ldots \sum_I \phi_3(S, I) \sum_D \phi_2(G, D, I) \psi_1(D)
\]

\[
\psi_2(G, I) \quad \psi_2(G, I, D)
\]
Recap: variable elimination (VE)

- marginalize over a subset - e.g., \[ P(J) = \sum_{C,D,I,G,S,L,H} P(C,D,I,G,S,L,J,H) \]
- expensive to calculate (why?)
- eliminate variables in some order (order of factors in the summation)
Recap: variable elimination (VE)

- eliminate variables in some order
- creates a chordal graph
- maximal cliques are factors created during VE ($\psi_t$)

$P(J)^?$

order:
C,D,I,H,G,S,L
Clique-tree

- summarize the VE computation using a clique-tree

1: $C, D$  \rightarrow_D  2: $G, I, D$  \rightarrow_{G,I}  3: $G, S, I$

4: $G, J$  \rightarrow_{G,J}  5: $G, J, S, L$  \rightarrow_{J,S,L}  6: $J, S, L$  \rightarrow_{J,L}  7: $J, L$

order: $C, D, I, H, G, S, L$

- clusters are maximal cliques (scope of factors created during VE)

$$C_i = \text{Scope}[\psi_i]$$
Clique-tree

- summarize the *VE computation* using a **clique-tree**

- **clusters** are maximal cliques (factors that are marginalized)

- **sepsets** are the result of marginalization over cliques

\[
C_i = \text{Scope}[\psi_i]
\]
\[
S_{i,j} = \text{Scope}[\psi'_i]
\]
\[
S_{i,j} = C_i \cap C_j
\]
Clique-tree: properties

- a tree $\mathcal{T}$ from clusters $C_i$ and sepsets $S_{i,j} = C_i \cap C_j$
- **family-preserving property:** $\alpha(\phi) = j$
  - each factor $\phi$ is associated with a cluster $C_j$ s.t. $\text{Scope}[\phi] \subseteq C_j$
Clique-tree: properties

- a tree $\mathcal{T}$ from clusters $C_i$ and sepsets $S_{i,j} = C_i \cap C_j$

  - **family-preserving property:** $\alpha(\phi) = j$
    - each factor $\phi$ is associated with a cluster $C_j$ s.t. $\text{Scope}[\phi] \subseteq C_j$

  - **running intersection property:**
    - if $X \in C_i, C_j$ then $X \in C_k$ for $C_k$ in the path $C_i \rightarrow \ldots \rightarrow C_j$
VE as message passing

think of VE as sending messages
**VE as message passing**

think of VE as sending messages

![Graph showing message passing]

- **Step 1:** $C, D$
  - $\delta_{1\rightarrow 2}(D) = \sum_C \psi_1(C_1)$
- **Step 2:** $G, I, D$
  - $\delta_{2\rightarrow 3}(G,I) = \sum_D \psi_2(C_2) \times \delta_{1\rightarrow 2}$
- **Step 3:** $G, S, I$
  - $\delta_{3\rightarrow 5}(G,S) = \sum_I \psi_3(C_3) \times \delta_{2\rightarrow 3}$
- **Step 4:** $H, G, J$
- **Step 5:** $G, J, S, L$

calculate the product of factors in each clique

\[ \psi_i(C_i) \triangleq \prod_{\phi: \alpha(\phi) = i} \phi \]

send messages from the leaves towards a root:

\[ \delta_{i\rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in \text{Nb}_{i-j}} \delta_{k\rightarrow i}(S_{i,k}) \]

neighbors
message passing

• think of VE as sending messages

\[
\delta(S_i) = \sum_{i \to j} \psi(C_i) \prod_{k \in N_{i \to j}} \delta_k \to j(S_{i,k})
\]

• send messages from the leaves towards a root:

\[
\delta_{i \to j}(S_{i,j}) = \sum_{C_i \sim S_{i,j}} \psi_i(C_i) \prod_{k \in N_{i \to j}} \delta_k \to j(S_{i,k})
\]

• the message is the marginal from one side of the tree
message passing

- think of VE as sending messages

- send messages from the leaves towards a root:
  \[ \delta_{i \to j}(S_{i,j}) \triangleq \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in N_{b_i} - j} \delta_{k \to i}(S_{i,k}) \]

- the belief at the root clique is
  \[ \beta_r(C_r) \triangleq \psi_r(C_r) \prod_{k \in N_{b_r}} \delta_{k \to r}(S_{r,k}) \]
  proportional to the marginal
  \[ \beta_r(C_r) \propto \sum_{X - C_i} P(X) \]
message passing: root-to-leaves

• what if we continue sending messages?

(from the root to leaves)

• clique $i$ sends a message to clique $j$ when received messages from all the other neighbors $k$
message passing: root-to-leaves

- what if we continue sending messages? (from the root to leaves)

- sum-product belief propagation (BP)

\[
\delta_{i\rightarrow j}(S_{i,j}) = \sum_{C_i-S_{i,j}} \psi_i(C_i) \prod_{k \in N_{b_i-j}} \delta_{k\rightarrow i}(S_{i,k})
\]

\[
\mu_{i,j}(S_{i,j}) \triangleq \delta_{i\rightarrow j}(S_{i,j})\delta_{j\rightarrow i}(S_{i,j})
\]

\[
\beta_i(C_i) \triangleq \psi_i(C_i) \prod_{k \in N_{b_i}} \delta_{k\rightarrow i}(S_{i,k})
\]

for any clique (not only root)
Summery so far...

- VE creates a chordal induced graph
- maximum cliques in this graph: clusters
- message passing view of VE:
  - send messages between clusters towards a root
- going beyond VE:
  - send messages back from the root
  - produce marginal over all clusters
Clique-tree: calibration

represent $P$ using marginals:

$$
\frac{\prod_i \beta_i}{\prod_{i,j \in E} \mu_{ij}} = \frac{\prod_i \psi_i \prod_{k \to i} \delta_{k \to i}}{\prod_{i,j \in E} \delta_{i \to j} \delta_{j \to i}} = \prod_i \psi_i = \tilde{P}
$$
Clique-tree: calibration

represent P using marginals: \[
\frac{\prod_i \beta_i}{\prod_{i,j \in E} \mu_{i,j}} = \frac{\prod_i \psi_i \prod_{k \rightarrow i} \delta_{k \rightarrow i}}{\prod_{i,j \in E} \delta_{i \rightarrow j} \delta_{j \rightarrow i}} = \prod_i \psi_i = \tilde{P}
\]

an arbitrary assignment to all \( \beta_i, \mu_{i,j} \) is calibrated iff 

\[\mu_{i,j}(S_{i,j}) = \sum_{C_i-S_{i,j}} \beta_i(C_i) = \sum_{C_j-S_{i,j}} \beta_j(C_j)\]
Clique-tree: calibration

represent $P$ using marginals: $$\frac{\prod_i \beta_i}{\prod_{i,j \in E} \mu_{i,j}} = \frac{\prod_i \psi_i \prod_{k \rightarrow i} \delta_{k \rightarrow i}}{\prod_{i,j \in E} \delta_{i \rightarrow j} \delta_{j \rightarrow i}} = \prod_i \psi_i = \tilde{P}$$

an arbitrary assignment to all $\beta_i, \mu_{i,j}$ is calibrated iff

$BP$ produces calibrated beliefs

$$\mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$$

being calibrated and $\star$ means that all $\beta_i, \mu_{i,j}$ are marginals

$$\tilde{P}(X) \propto \frac{\prod_i \beta_i(C_i)}{\prod_{i,j \in E} \mu_{i,j}(S_{i,j})} \iff \beta_i(C_i) \propto P(C_i)$$
BP: an alternative update

**approach 1. message update**

\[
\delta_{i\rightarrow j}(S_{i,j}) = \sum_{C_i = S_{i,j}} \psi_i(C_i) \prod_{k \in N_{bi} - j} \delta_{k\rightarrow i}(S_{i,k})
\]

calculate the beliefs **in the end**

\[
\beta_i(C_i) = \psi_i(C_i) \prod_{k \in N_{bi}} \delta_{k\rightarrow i}(S_{i,k})
\]

Update the beliefs so that:

- they are calibrated
- they satisfy
BP: an alternative update

**approach 1. message update**

\[
\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i(C_i) \prod_{k \in N_{b_i} - j} \delta_{k \rightarrow i}(S_{i,k})
\]

calculate the beliefs in the end

\[
\beta_i(C_i) = \psi_i(C_i) \prod_{k \in N_{b_i}} \delta_{k \rightarrow i}(S_{i,k})
\]

**approach 2. belief update idea**

Update the beliefs so that:

- they are calibrated
  \[
  \mu_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)
  \]

- they satisfy
  \[
  \frac{\prod_i \beta_i}{\prod_{i,j \in \mu_{i,j}} \mu_{i,j}} = \prod_i \psi_i
  \]
BP: an alternative update

belief update

initialize $\beta_i \leftarrow \psi_i = \prod_{\phi: \alpha(\phi) = i} \phi$, $\mu_{i,j} \leftarrow 1$

until convergence:

pick some $(i, j) \in \mathcal{E}$

$$\hat{\mu}_{i,j} \leftarrow \sum_{C_i - S_{i,j}} \beta_i$$  // $\hat{\mu}_{i,j} = \delta^{\text{new}}_{i \rightarrow j} \delta_{j \rightarrow i}$

$$\beta_j \leftarrow \beta_j \frac{\hat{\mu}_{i,j}}{\mu_{i,j}}$$  // $\frac{\hat{\mu}_{i,j}}{\mu_{i,j}} = \frac{\delta^{\text{new}}_{i \rightarrow j} \delta_{j \rightarrow i}}{\delta^{\text{old}}_{i \rightarrow j} \delta_{j \rightarrow i}} = \frac{\delta^{\text{new}}_{i \rightarrow j}}{\delta^{\text{old}}_{i \rightarrow j}}$

$$\mu_{i,j} \leftarrow \hat{\mu}_{i,j}$$

at convergence, beliefs are calibrated and so they are $\propto$ marginals

$$\sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j)$$
What **type of queries** can we answer?

- **marginals** over subset of cliques $P(A) \quad A \subseteq C_i$
What **type of queries** can we answer?

- **marginals** over subset of cliques \( P(A \mid A \subseteq C_i) \)
- updating the beliefs after **new evidence** \( P(A \mid E^{(t)} = e^{(t)}) \mid A \subseteq C_i, E \subseteq C_j \)
  - multiply the *(previously calibrated)* beliefs \( \beta(C_i)(E^{(t)} = e^{(t)}) \)
  - propagate to recalibrate *(belief update procedure)*
Clique-tree & queries

What **type of queries** can we answer?

- **marginals** over subset of cliques \( P(A) \ A \subseteq C_i \)
- updating the beliefs after **new evidence** \( P(A \mid E^{(t)} = e^{(t)}) \ A \subseteq C_i, E \subseteq C_j \)
  - multiply the *(previously calibrated)* beliefs \( \beta(C_i) \mathbb{1}(E^{(t)} = e^{(t)}) \)
  - propagate to recalibrate *(belief update procedure)*
- **marginals outside cliques**: \( P(A, B) \ A \subseteq C_i, B \subseteq C_j \)
  - define a super-clique that has both A,B
  - a more efficient alternative?
Clique-tree & queries

What type of queries can we answer?

- marginals over subset of cliques: $P(A) \quad A \subseteq C_i$
- updating the beliefs after new evidence: $P(A \mid E^{(t)} = e^{(t)}) \quad A \subseteq C_i, E \subseteq C_j$
  - multiply the (previously calibrated) beliefs: $\beta(C_i)\mathbb{I}(E^{(t)} = e^{(t)})$
  - propagate to recalibrate (belief update procedure)
- marginals outside cliques: $P(A, B) \quad A \subseteq C_i, B \subseteq C_j$
  - define a super-clique that has both A,B
  - a more efficient alternative?
- partition function: $Z = \sum_{C_i} \beta_i(C_i)$
Building a clique-tree

how to create it for a given graphical model:
Building a clique-tree

how to create it for a given graphical model:

1. triangulation: make a chordal graph
   - e.g. induced graph in VE
   - finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)
Building a clique-tree

how to create it for a given graphical model:

1. triangulation: make a chordal graph
   - e.g. induced graph in VE
   - finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)

2. find maximal cliques (clusters in the clique-tree)
   - in general graphs NP-hard, but easy for chordal graphs
   - assign each factor to a clique

image: wikipedia
Building a clique-tree

how to create it for a given graphical model:

1. **triangulation:** make a chordal graph
   - e.g. *induced graph* in VE
   - finding the chordal graph with min max-clique is NP-hard (heuristics we discussed)

2. find maximal cliques (**clusters** in the clique-tree)
   - in general graphs NP-hard, but easy for chordal graphs
   - assign each factor to a clique

3. use max. spanning-tree to build a tree (edge-cost $|C_i \cap C_j|$)
   
   image: wikipedia
Building a clique-tree: example

input

from: wainwright & jordan
Building a clique-tree: example

input

1  2  3
4  5  6
7  8  9

triangulated

1  2  3
4  5  6
7  8  9

from: wainwright & jordan
Building a clique-tree: example

input

triangulated

clique-tree

from: wainwright & jordan
clique-tree quiz

what clique-tree to use here?
what are the sepsets?
cost of exact inference?
Summary

• **VE as message passing** in a clique-tree
• **clique-tree**: running intersection & family preserving
• **belief propagation** updates:
  ▪ message update
  ▪ belief update
• **types** of queries
• how to **build** a clique-tree for exact inference
Chordal graph and clique-tree

Chordal graph = Markov ∩ Bayesian networks

convert MRF to Bayes-net (the actual procedure):

- triangulate
- build a clique-tree
- within cliques: fully connected directed edges
- between cliques: from root to leaves