Probabilistic Graphical Models

Raquel Urtasun and Tamir Hazan

TTI Chicago

April 25, 2011

Clique Trees

- Today we are going to see an alternative implementation of the same insight as VE.
- We define a set of factors Φ over a set of variables $\mathcal{X}$, where each factor ϕ_i has a scope $\mathbf{X}_i$.
- The set of factors defines the unnormalized distribution

$$\hat{P}_\Phi(\mathcal{X}) = \prod_{\phi_i \in \Phi} \phi_i(\mathbf{X}_i)$$

- In BN the factors are CPD and the distribution is normalized.
- In BN when dealing with evidence $\mathbf{E} = \mathbf{e}$, the probability is $\hat{P}_\Phi(\mathcal{X}) = P_\mathcal{B}(\mathcal{X}, \mathbf{e})$.
- For a Markov network, then $\hat{P}_\Phi(\mathcal{X}) = \hat{P}_\Phi$, the unnormalized Gibbs measure.

VE and Clique Trees

- For VE, we multiply factors to get ψ_i, and we sum the to get a new factor τ_i.
- Different view of this process, where
  - A factor ψ_i is a **computational data structure**,
  - which takes messages τ_j generated by other factors ψ_j,
  - and generates messages τ_i,
  - which are used by another factor ψ_l.

Cluster Graphs and Family Preserving Property

- A **cluster graph** is a data structure that provides a graphical flowchart of the process of manipulating the factors.
- Each node in the cluster graph is a **cluster**, which is associated with a subset of variables.
- The graph contains undirected edges that connect clusters which scopes have non-empty intersections.
- **Def:** A cluster graph $\mathcal{U}$ for a set of factors Φ over $\mathcal{X}$ is an undirected graph, with nodes i associated with a subset $\mathbf{C}_i \subseteq \mathcal{X}$. A cluster graph must be **family preserving**, each factor $\phi \in \Phi$ must be associated with a cluster $\mathbf{C}$, denoted $\alpha(\phi)$, such that $Scope(\phi) \subseteq \mathbf{C}_i$. Each edge between a pair of clusters $\mathbf{C}_i$ and $\mathbf{C}_j$ is associated with a sepset $\mathbf{S}_{i,j} = \mathbf{C}_i \cap \mathbf{C}_j$.

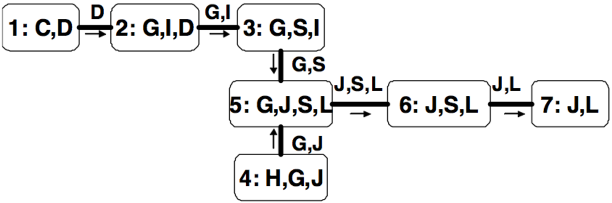
Cluster graphs and VE

- An execution of VE defines a cluster graph.
- A cluster for each factor ψ_i, which is associated with the set of variables $\mathbf{C}_i = Scope(\psi_i)$.
- There is an edge between $\mathbf{C}_i$ and $\mathbf{C}_j$ if the "message" τ_i produced by eliminating a variable in ψ_i is used in the computation of τ_j.

Example

Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$, $\phi_D(D,C)$	C, D	$\tau_1(D)$
2	D	$\phi_G(G,I,D)$, $\tau_1(D)$	G, I, D	$\tau_2(G,I)$
3	I	$\phi_I(I)$, $\phi_S(S,I)$, $\tau_2(G,I)$	G, S, I	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	H, G, J	$\tau_4(G,J)$
5	G	$\tau_4(G,J)$, $\tau_3(G,S)$, $\phi_L(L,G)$	G, J, L, S	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S)$, $\phi_J(J,L,S)$	J, L, S	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	J, L	$\tau_7(J)$

$(P(J))$



(Clique Tree)

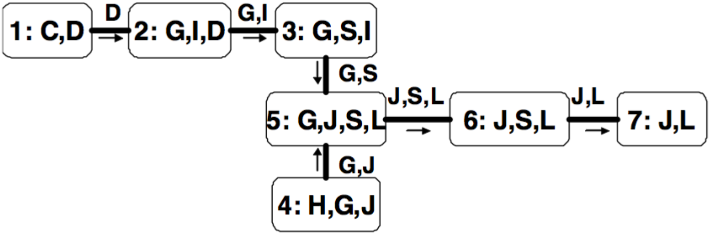
Properties of VE Cluster Graph

The cluster graph of VE has certain properties:

- The cluster graph induced by the execution of VE is a tree, as it uses each intermediate factor τ_i at most once.
- The same for ϕ_i, it is used once to create ψ_j and removed from the set of factors Φ.
- The cluster graph is defined to be undirected, however an execution of VE gives directionality.
- The messages induced a directed tree, with all the messages flowing towards a single cluster where the final result is computed.
- This cluster is called the **root** of the directed tree.
- If $\mathbf{C}_i$ is on the path of $\mathbf{C}_j$ to the root, then $\mathbf{C}_i$ is **upstream** from $\mathbf{C}_j$ and C_j is **downstream** from $\mathbf{C}_i$.

Running intersection property

Def: Let $\mathcal{T}$ be a cluster tree over Φ, with $\mathcal{V}_\mathcal{T}$ its vertices and $\mathcal{E}_\mathcal{T}$ its edges. $\mathcal{T}$ has the **running intersection** property if, whenever $X \in \mathbf{C}_i$ and $X \in \mathbf{C}_j$, then X is also in every cluster in the (unique) path in $\mathcal{T}$ between $\mathbf{C}_i$ and $\mathbf{C}_j$.



Intuition: This holds in cluster trees induced by VE because a variable appears in every factor from the moment it is introduced until it is eliminated.

Running intersection property

Theorem: Let $\mathcal{T}$ be a cluster tree induced by VE over Φ. Then $\mathcal{T}$ satisfies the running intersection property.

- Proof: Let $\mathbf{C}$ and $\mathbf{C}'$ be two clusters containing X, and let $\mathbf{C}_X$ the cluster where X is eliminated.
- We need to prove that X must be in every cluster on the path from $\mathbf{C}$ to $\mathbf{C}_X$ (and the same for $\mathbf{C}'$).
- The computation of $\mathbf{C}_X$ is later in the algorithm than $\mathbf{C}$, as after elimination there is no more factor containing that variable.
- By assumption X is in the domain of $\mathbf{C}$, and X is not eliminated in $\mathbf{C}$.
- Therefore the message computed in $\mathbf{C}$ must have X in its domain.
- By definition the neighbors upstream in the tree of $\mathbf{C}$ multiply in the message from $\mathbf{C}$, so it's in the scope.
- We can apply the same argument to traverse upstream until $\mathbf{C}_X$ where the node is eliminated.
- Thus $\mathbf{X}$ appears in all the cliques between $\mathbf{C}$ ($\mathbf{C}'$) and $\mathbf{C}_X$.

More on Running intersection property

Theorem: Let $\mathbf{C}$ and $\mathbf{C}'$ be two clusters containing X, and let $\mathbf{C}_i$ and $\mathbf{C}_j$ be two neighboring clusters, such that $\mathbf{C}_i$ passes the message τ_i to $\mathbf{C}_j$. Then $Scope(\tau_i) = \mathbf{S}_{i,j}$.

- Similar argument as the previous theorem.
- A cluster tree that satisfies the running intersection property is very useful for exact inference in graphical models.

Even more on Running intersection property

Def: Let Φ be a set of factors over $\mathcal{X}$. A cluster tree over Φ that satisfies the running intersection property is called a **clique tree**. In the case of a clique tree, the clusters are called **cliques**.

- Remember that by definition a cluster tree satisfy the family preserving property: each factor is associated with a $\mathbf{C}_i$ and each edge between $\mathbf{C}_i$ and $\mathbf{C}_j$ is associated with the sepset $\mathbf{S}_{i,j}$.
- This definition is equivalent to say that $\mathcal{T}$ is a clique tree for Φ iff it is a clique tree for a chordal graph containing $\mathcal{H}_\Phi$.
- These properties are true iff the clique tree admits VE by message passing over the tree.

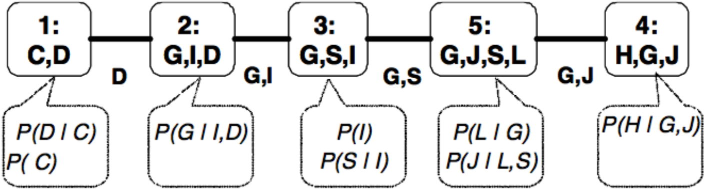
Clique Tree and Separation set

- Running intersection property implies independence.
- Let $\mathcal{T}$ be a cluster tree over Φ, and let $\mathcal{H}_\Phi$ be the undirected graph associated with this factors.
- **Theorem:** $\mathcal{T}$ satisfies the running intersection property iff, for every sepset $\mathbf{S}_{i,j}$ we have that $\mathbf{W}_{<(i,j)}$ and $\mathbf{W}_{<,(j,i)}$ are separated in $\mathcal{H}_\Phi$ given $\mathbf{S}_{i,j}$.
- Moreover, each node in $\mathcal{T}$ corresponds to a clique in a chordal graph $\mathcal{H}'$ containing $\mathcal{H}$, and each maximal clique in $\mathcal{H}'$ is represented in $\mathcal{T}$.

Message passing: Sum product

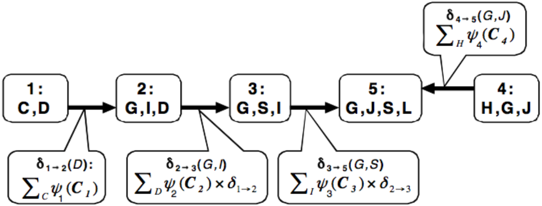
- Given an clique tree, it can be used as the basis for many different VE executions.
- Provides a structure for catching computations, so that multiple executions of variable elimination can be performed more efficiently than doing each one separately.
- Given a clique tree $\mathcal{T}$, it is guaranteed to satisfy the family preservation and running intersection property.
- This is because it's a cluster graph, and we prove the running intersection property for clique trees.
- If clique $\mathbf{C}'$ requires a message from $\mathbf{C}$, then $\mathbf{C}'$ must wait until $\mathbf{C}$ performs the computation and sends the message.

Example



- $\mathcal{T}$ satisfies the family preservation and the running intersection property.
- We have specified the assignment α of the initial factors to cliques.
- We might have more than one choice.
- First task is to compute the initial potentials $\psi_i(\mathbf{C}_i)$ by multiplying the initial factors assigned to the clique $\mathbf{C}_i$.
- $P(J)$?

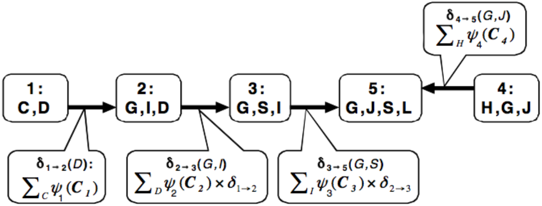
Message propagations



($\mathbf{C}_5$ is the root)

- The operations could also have been done in another order.
- The only constraint is that a clique gets all of its incoming messages from its downstream neighbors before it sends its outgoing messages to the upstream neighbor.
- A clique is **ready** when it has received all of its incoming messages.
- Is $\{\mathbf{C}_1, \mathbf{C}_4, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_5\}$ legal? And $\{\mathbf{C}_2, \mathbf{C}_1, \mathbf{C}_4, \mathbf{C}_3, \mathbf{C}_5\}$?

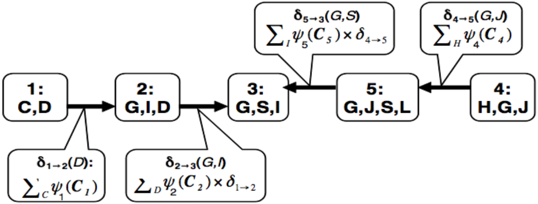
Choice of root



($\mathbf{C}_5$ is the root)

- The choice of root is not fully determined.
- To derive $P(J)$ we could have chosen C_4 as the root.
- What would be the execution?

Other variables



($\mathbf{C}_3$ is the root)

- What if we want to compute $P(G)$?
- What are the possible roots?
- Choose one where the variable appears, doesn't matter which one.
- To compute marginals of different variables we are reusing computation, e.g., $\mathbf{C}_1$ and $\mathbf{C}_2$.