

KRR - Clique Trees for Exact Inference in Graphical Models

Tudor Berariu, Alexandru Sorici

November 2018

1 Clique Trees and Belief Propagation

The goal of this tutorial is to briefly introduce *clique trees* and their use in performing exact inference for probabilistic graphical models. Our focus is on solving inference queries in Bayesian Networks using belief propagation, but the algorithm can be used with only minor modifications for Markov Networks also.

The current tutorial assumes the reader is familiar with the factor representation, and also knows how to multiply two factors, and how to sum out one or more variables from a factor.

Subsection 1.1 introduces the needed definitions followed by an algorithm for constructing clique trees for Bayesian Networks. Subsection 1.2 describes the belief propagation algorithm for clique trees.

1.1 Constructing clique trees for Bayesian Networks

Clique graphs. A *clique graph* \mathcal{C} is an undirected graph whose vertices are cliques of variables. Edges are also labelled with the variables from the intersection of the two clusters it connects.

$$\{A, B, C, D\} \overset{A,C}{-} \{A, C, E, F\} \overset{F}{-} \{F, G\}$$

Clique graphs. A subgraph of a clique graph \mathcal{C} is a *clique tree* (or *junction tree*) \mathcal{T} if:

1. \mathcal{T} is a spanning tree (it contains all the nodes in \mathcal{C});
2. it satisfies the *running intersection property*:
 - For any two cliques C_i, C_j (corresponding to some vertices in \mathcal{T}), all the cliques C_k from the path between C_i and C_j contain $C_i \cap C_j$.

Given a Bayesian Network with its associated directed acyclic graph \mathcal{G} a *clique tree* \mathcal{T} is constructed following the next steps:

1. Moralize the graph \mathcal{G} . Starting from the directed acyclic graph \mathcal{G} , create the undirected graph \mathcal{H} with the same set of vertices as \mathcal{G} . The moralized graph \mathcal{H} will have an edge between the nodes X, Y if (1) there is an arc between X and Y in \mathcal{G} , or (2) if X , and Y are both parents of some common node Z in \mathcal{G} .
2. Triangulate \mathcal{H} . Add necessary edges in \mathcal{H} such that there are no cycles of length higher than 3 without a chord.
 - There are many ways in which edges can be added in order to triangulate a graph. Some good heuristic is to *eliminate* one variable at a time such that the number of extra edges added at each step is minimal¹.
3. Create the clique graph \mathcal{C} . Find all the maximal cliques in \mathcal{H} and consider each a vertex in the new graph \mathcal{C} . Each two vertices are connected if their corresponding cliques share any variables. Each edge in \mathcal{C} is labelled with the intersection of the two cliques it connects in \mathcal{C} .
4. Create a clique tree \mathcal{T} from \mathcal{C} by applying some algorithm for finding maximum weight spanning trees (e.g. Kruskal's). The weight associated to an edge is given by the number of common variables between the two cliques $w_{ij} = |C_i \cap C_j|$.

¹<http://staff.utia.cas.cz/vomlel/55900566.pdf>

Each CPD for $P(X \mid \text{parents}(X))$ in the initial Bayesian Network is associated to that single node in \mathcal{T} that contains all the involved variables, i.e. X and its parents. In order to incorporate some evidence $\mathbf{Z} = z$ only the entries that match the observations are being kept. The clique potential for a node $V \in \mathcal{T}$ is the product of all those CPDs seen as factors.

$$\phi_C = \prod_{\phi: \text{var}(\phi) \subseteq C} \phi$$

1.2 Belief propagation

In what follows the *belief propagation* algorithm for exact inference is described.

Message Passing. Belief propagation builds on the idea of message passing between cliques. Passing a message between cliques C_i and C_j takes two steps: (1) projecting ϕ_{C_i} on $|C_i \cap C_j|$, and (2) absorbing the resulting potential in ϕ_{C_j} .

Belief propagation. Performing belief propagation in a clique tree takes four steps:

1. Incorporating the evidence $\mathbf{Z} = \mathbf{z}$: for each variable $Z_i \in \mathbf{Z}$ choose a single factor in \mathcal{T} that contains Z_i and set to zero all entries that do not match $Z_i = z_i$;
2. Choosing some root node $V \in \text{nodes}(\mathcal{T})$;
3. Propagating messages from the leafs to the root:
 - each leaf sends a message to its parent by summing out all variables except those it has in common with its parent;
 - each non-leaf node waits for a message from all its children, multiplies the received factor with its table, and then acts as a leaf (i.e. it sends a message to its parent);
 - the process ends once the root V received and absorbed the message from all its children;
4. Propagating messages from the root back to the leafs.
 - the root V divides its current table by the factor received by a child and projects the result to their common variables in order to send the marginalised factor back to the child;
 - each node waits for the message from its parent, multiplies its factor with the received message and then acts as the root (i.e. it sends messages further to its children);
 - the process ends once each leaf received and absorbed the returning message from its parent;

Result. The factors that result after *belief propagation* are unnormalized joint probabilities of the variables in each clique. These can be further used to compute any needed query.

2 Example

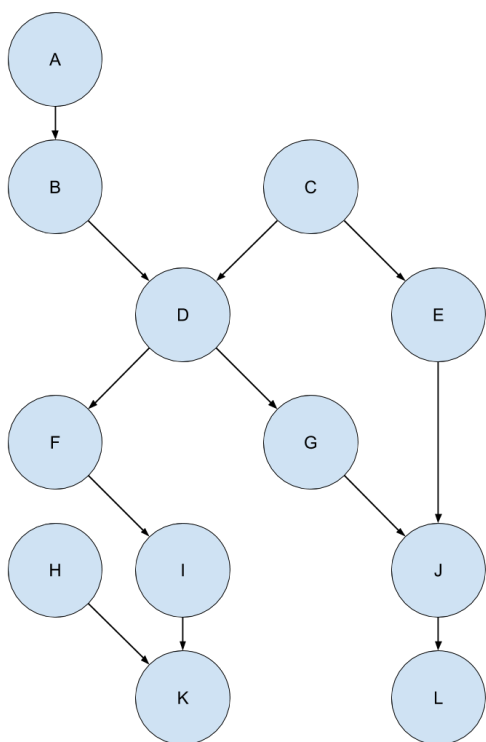
The input file. On each line in the input file the variable, its parents, and its CPD are separated by “;”. The values in the CPD on each line are ordered as in this case for variable D:

1. $P(D = 1 \mid B = 0, C = 0) = 0.1$
2. $P(D = 1 \mid B = 0, C = 1) = 0.5$
3. $P(D = 1 \mid B = 1, C = 0) = 0.15$
4. $P(D = 1 \mid B = 1, C = 1) = 0.9$

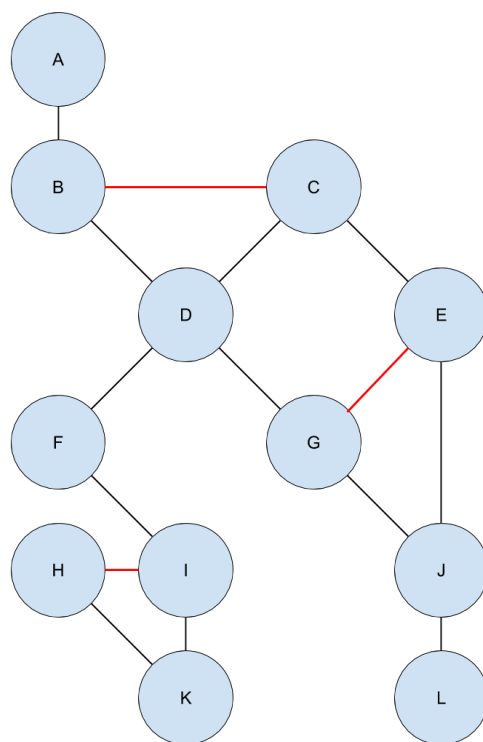
```

A ; ; 0.8
B ; A ; 0.8 0.1
C ; ; 0.9
D ; B C ; 0.1 0.5 0.15 0.9
E ; C ; 0.8 0.5
F ; D ; 0.9 0.45
G ; D ; 0.9 0.75
H ; ; 0.15
I ; F ; 0.75 0.45
J ; E G ; 0.25 0.15 0.25 0.15
K ; H I ; 0.25 0.15 0.5 0.75
L ; J ; 0.15 0.25

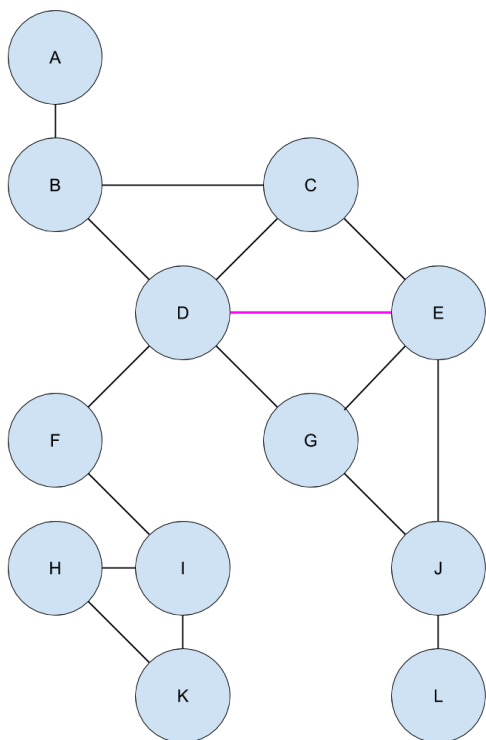
```



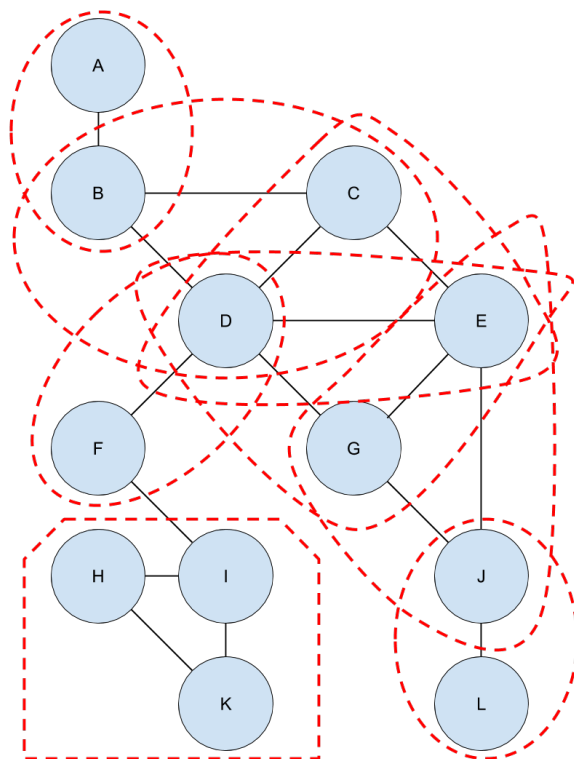
(a) The original Bayes Network



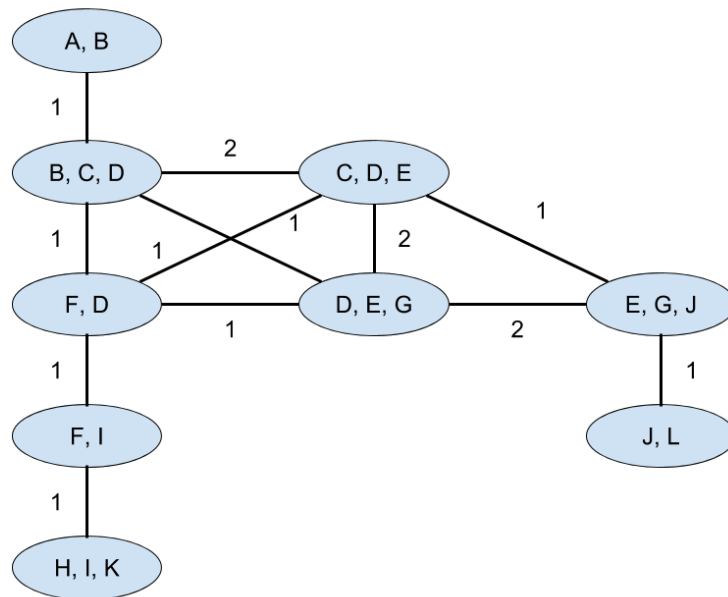
(b) The moralized graph



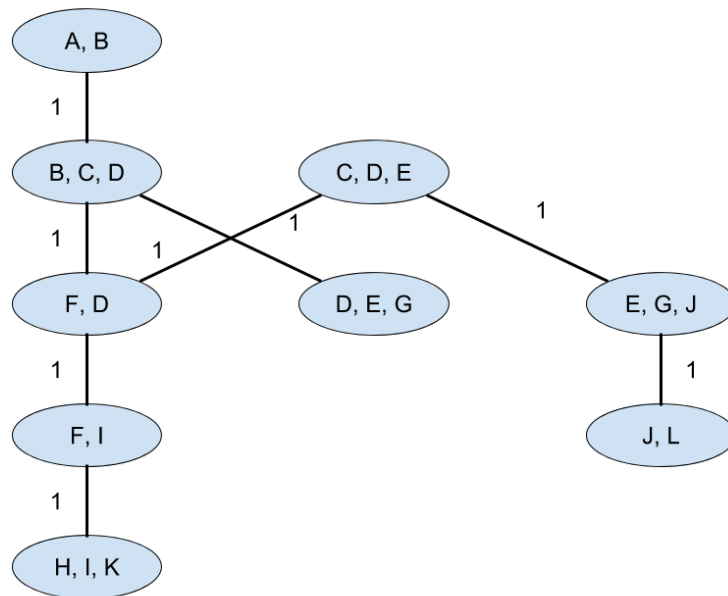
(c) Triangulation



(d) Maximal cliques



(a) The clique graph



(b) A possible clique tree (a minimum spanning tree of \mathcal{C})