CONDITIONAL INDEPENDENCE AND D-SEPARATION

MOHAMMAD REZA KARIMI

Markovian Assumption. We first remind the reader of the *Markovian* assumption on Bayesian networks. Whenever one draws a DAG representing a Bayesian network, this assumption is always assumed: for every node V, we have

$$V \perp \text{non-descendants}(V) \,|\, \text{parents}(V).$$

Another Algorithm for D-separation. Here we bring another (efficient) algorithm for d-separation and prove its correctness. Let G be a DAG and $\boldsymbol{X}, \boldsymbol{Y}$, and $\boldsymbol{Z}$ be subsets of nodes of G. The goal is to decide whether $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$. The following algorithm creates a subgraph G' as follows:

(1) For every leaf[1] W, remove W if it is not in $\boldsymbol{X}, \boldsymbol{Y}$, or $\boldsymbol{Z}$. Repeat this step until no leaf can be removed.
(2) Delete all outgoing edges of vertices in $\boldsymbol{Z}$.

Now decide $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ only if they are disconnected in G'.

We now prove the correctness of this algorithm. The proof has two parts:
Step 1. Suppose that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ (in G). This means that every path between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is blocked by $\boldsymbol{Z}$ in G. We have to show that any such path is nonexistent in G': either a node or an edge is deleted from every such path.

Let α be a path from $\boldsymbol{X}$ to $\boldsymbol{Y}$ in G that is blocked by $\boldsymbol{Z}$. One of the following cases should happen:

- There is a node $T \in \boldsymbol{Z}$ in the path such that the sequence of edges looks like $\to T \to$, $\leftarrow T \leftarrow$, or $\leftarrow T \to$ on the path. In all of these cases, T has at least one outgoing edge that is deleted by our algorithm. Thus, this path would not remain in G'.
- There is a node T with the sequence $\to T \leftarrow$, such that T and none of its descendants are in $\boldsymbol{Z}$ (thus making the path blocked). Assuming that none of T's descendants are in $\boldsymbol{X}$ or $\boldsymbol{Y}$, T will be pruned in our algorithm. If T has a descendant S in *e.g.* $\boldsymbol{Y}$, then the path from $\boldsymbol{X}$ to T and from T to S will have at least one $\to \cdot \leftarrow$ less. Continuing this procedure results in a path that is in the former case, or a node that has no descendant in $\boldsymbol{X}$ or $\boldsymbol{Y}$.

E-mail address: `mkarimi@inf.ethz.ch`.
Date: 4 October 2019.
[1] A *leaf* is a node without any descendants.

Step 2. Suppose that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *not* d-separated by $\boldsymbol{Z}$. We show that any non-blocked path connecting $\boldsymbol{X}$ and $\boldsymbol{Y}$ in G is still remaining in G'. Take such a path. For any node in the path, we have the following cases:

- The node is $\rightarrow T \leftarrow$. By assumption, T or one of its descendants should be in $\boldsymbol{Z}$, which prevents T to be pruned by the algorithm.
- The node is one of the other three types. Following on the outgoing direction of the node, we either reach a node in $\boldsymbol{X}$ or $\boldsymbol{Y}$, or we reach some node $\rightarrow S \leftarrow$, which by the reasoning above, should have a descendant in $\boldsymbol{Z}$. Hence, this node is not pruned.

Exercise. Let X and Y be two nodes in a DAG G that are not connected by an edge. Let $\boldsymbol{Z}$ be a set of nodes defined as follows: $Z \in \boldsymbol{Z}$ if and only if $Z \notin \{X, Y\}$ and Z is an ancestor of X or an ancestor of Y. Show that X and Y are d-separated by $\boldsymbol{Z}$.
I strongly recommend to prove this exercise yourself before proceeding to the proof!

Proof. Let α be a path between X and Y. We show that α is blocked by $\boldsymbol{Z}$. If the edge immediately after X is $X \leftarrow T$, then T is an ancestor of X and is in $\boldsymbol{Z}$, hence blocks α. With the same argument for Y, we derive that α looks like $X \rightarrow \cdots \leftarrow Y$. By continuing the path from X, there should be a first node T such that the direction of the edges change. Also continuing the path from Y, there should be a first time that the direction of edges change, call it S. That is,

$$\alpha = X \rightarrow \cdots \rightarrow T \leftarrow \cdots ? \cdots \rightarrow S \leftarrow \cdots \leftarrow Y.$$

We first note that T is not an ancestor of X: if it was, then the graph had a cycle ($X \rightarrow \cdots T \rightarrow \cdots X$). With the same argument, none of T's descendants are ancestors of X. Similarly, S and none of its descendants are ancestors of Y.

For the path to be active, both S and T (or one of their descendants) should be in $\boldsymbol{Z}$. This can only be achieved if T (or one of its descendants) is an ancestor of Y *and* S (or one of its descendants) be an ancestor of X. But this makes the following cycle in the graph:

$$X \rightarrow \cdots T \rightarrow \cdots Y \rightarrow \cdots S \rightarrow \cdots X,$$

which is a contradiction. Hence, X and Y are d-separated by $\boldsymbol{Z}$. $\square$

Note. This discussion about the other proof for "Random Walk on Graphs" at the end of the tutorial session is non-examinable, so we do not bring it here.