

Probabilistic Artificial Intelligence

Problem Set 2

Oct 13, 2017

1. Bayesian Networks: d-separation

As discussed in class, conditional independence properties entailed by a Bayesian Network can be read directly from the graph, using the notion of *d-separation*. Given the Bayesian network in Figure 1, which of the following conditional independence statements hold:

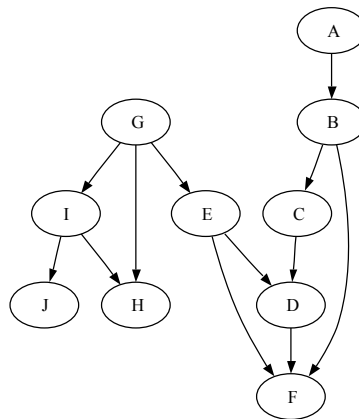


Figure 1: Bayes' net for Problem 1

1. $A \perp F$
2. $A \perp G$
3. $B \perp I \mid F$
4. $D \perp J \mid G, H$
5. $I \perp B \mid H$
6. $J \perp D$
7. $I \perp C \mid H, F$

2. Bayesian Networks: Variable elimination

Suppose you wish to do variable elimination on the Bayesian Network in Figure 1. Consider the variable ordering $A, B, C, D, E, F, G, H, I, J$. For each iteration of the algorithm (i.e., for each of the variables), determine which factors are removed and introduced. As an example, in the first iteration (variable A), the factor $P(A)$ and $P(B \mid A)$ are removed, and replaced by a new factor $g_1(B)$.

3. An algorithm for d-separation

In this exercise, you will implement an algorithm for computing d-separation of variables in Bayesian networks. In fact, the algorithm works in the opposite way by finding all variables that are *not* d-separated from a variable of interest. More formally, given a query variable X and a set of observed variables $\mathcal{Y} = \{Y_1, \dots, Y_n\}$, the algorithm decides which variables could be dependent on X given \mathcal{Y} , that is, it returns the set of reachable variables

$$\mathcal{R}(X \mid \mathcal{Y}) = \{Z \mid Z \notin \text{d-sep}(X; Z \mid \mathcal{Y})\}.$$

You are provided some skeleton Python code in the .zip file accompanying this document. Take the following steps for this exercise.

1. Install the Python dependencies listed in `README.txt`, if your system does not already satisfy them. After that, you should be able to run `demo.py` and produce some plots, albeit wrong ones for now.
2. Implement the missing code in `core.py` marked with `TODO`.
3. If your implementation is correct, you should get correct results for the v-structure of the demo file. You can try out more example networks from `examples_dsep.py` to test your implementation.
4. Now, create the network shown in [Figure 2](#) and answer the following questions:
 - (a) Which variables are reachable from Radio, if nothing is observed?
 - (b) Which variables are reachable from Radio, if Phone is observed?
 - (c) Which variables are reachable from Radio, if both Phone and Earthquake are observed?

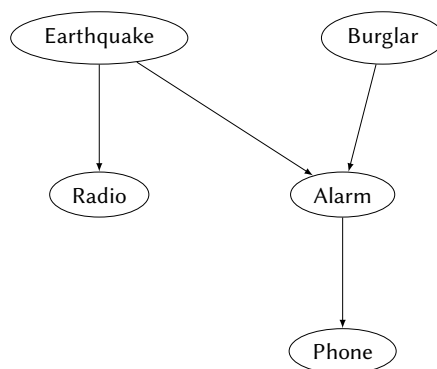


Figure 2: The earthquake network to be implemented.