Probabilistic Foundations of Artificial Intelligence

Problem Set 3

Oct 17, 2014

1. 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 $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$, 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 1 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?

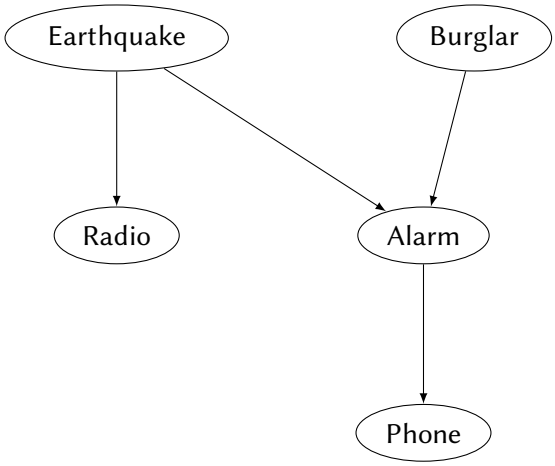

Figure 1: The earthquake network to be implemented.