# Bayesian Network Library

**Ameer Khan**

The Bayesian Network class inherits from the DirectedGraph class from graph.py (They should be in the same directory). evidenceList is the list of tuple pairs, specifying nodes on which evidence is set, and the state values.

A Bayesian Network object is instantiated as follows:

```
import bayesian;

networkA = bayesian.BayesianNet( );
```

Nodes can be added to the network using the addNode( ) method, and edges using the addChild( ) method. Note that there will be no action if a duplicate node is added using addNode( ) or addChild( ). addChild( ) will create the two specified nodes to the graph if they do not exist already, and connect them with a directed edge.

```
networkA.addNode( 'Smoking' );
networkA.addChild( 'Smoking', 'Cancer' );
```

Probability information is added to a node using the addProbabilityTable( ) method. The network must be created before calling this method on the each node. The probability specified must be either the marginal probability table for independent nodes, and conditional probability table for dependent nodes. The dependency list is also to be specified for dependent nodes, which is in the same order as the dimensions of the conditional probability table.

```
networkA.addProbabilityTable( 'Smoking', np.array( [ 0.8, 0.15, 0.05 ] ),
                              [ 'None', 'Light', 'Heavy' ] );
networkA.addProbabilityTable( 'Cancer', np.array( [ [ 0.96, 0.88, 0.60 ],
                                                    [ 0.03, 0.08, 0.25 ],
                                                    [ 0.01, 0.04, 0.15 ] ] ),
                              ['None', 'Benign', 'Malignant'], ['Smoking'] );
```

Joint probability of the network is computed using jointProbability( ). If a vertex list is provided, the joint probability is computed only for the given nodes. If the evidence in the network is set, the vertex list is ignored, and the joint probability of the entire network is computed. The corresponding dimensions of the joint probability are also collapsed when evidence is set on them.

```
prob = networkA.jointProbability( );
```

marginalProbability( ) computes the marginal probability for a specified node using the joint probability. If total is false, only the joint probability of the node and its parents are used to compute the marginal. If it is true, the joint probability of the full network is used. If evidence is set, total is set to true regardless of its specified value.

```
marginalProb = networkA.marginalProbability( 'Cancer' );
```

setEvidence( ) sets evidence in the network. The evidence list is provided as a list of tuple pairs containing the node to be set and the state value. unsetEvidence( ) clears the evidence set on the network.

```
networkA.setEvidence( [ ( 'Smoking', 'Light' ) ] );
```

getInference( ) gets the inference on a specified node based on the evidence set on the network.

```
cancerInference = networkA.getInference( 'Cancer' );
```