

Bayesian Networks Lab

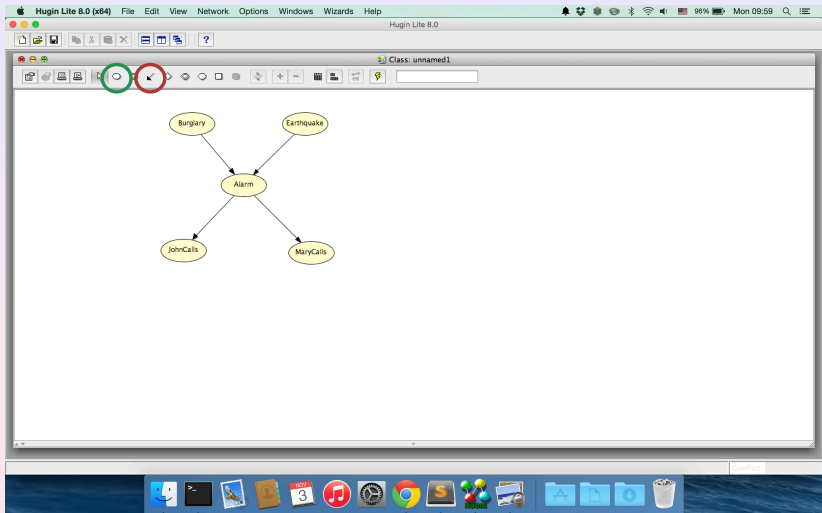
Andrea Passerini

Machine Learning

HuginLite

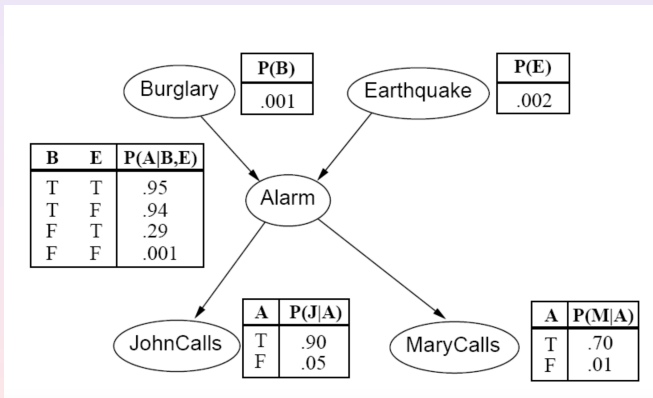
- Trial version of the Hugin family of software for Bayesian Networks
- The free trial version is limited to handle max. 50 states and learn from max. 500 cases
- It is prohibited to use the free Hugin Lite for any other purpose than the demonstration of capabilities and proof of concept
- Freely available at
<http://www.hugin.com/index.php/hugin-lite/>

Defining Nodes and Links



Defining the States

- Open CPT by clicking on a node holding the CTRL key
- Rename states, insert probability for each configuration



Compiling the Network

The screenshot displays the Hugin Lite 8.0 interface. At the top, the menu bar includes File, Edit, View, Network, Options, Windows, Wizards, and Help. The main window shows a Bayesian network diagram with five nodes: Burglary, Earthquake, Alarm, JohnCalls, and MaryCalls. Arrows indicate dependencies: Burglary and Earthquake are parents of Alarm, and Alarm is the parent of both JohnCalls and MaryCalls.

Below the diagram, a probability table is shown for the Alarm node. The table has columns for Burglary, Earthquake, Alarm, JohnCalls, and MaryCalls. The rows represent the states of the parent nodes (Burglary and Earthquake).

Earthquake...	Burglary...	Alarm	JohnCalls	MaryCalls
yes	yes	0.29	0.94	0.001
no	yes	0.71	0.06	0.999

The interface also includes a toolbar with various icons, a status bar at the bottom, and a taskbar with application icons.

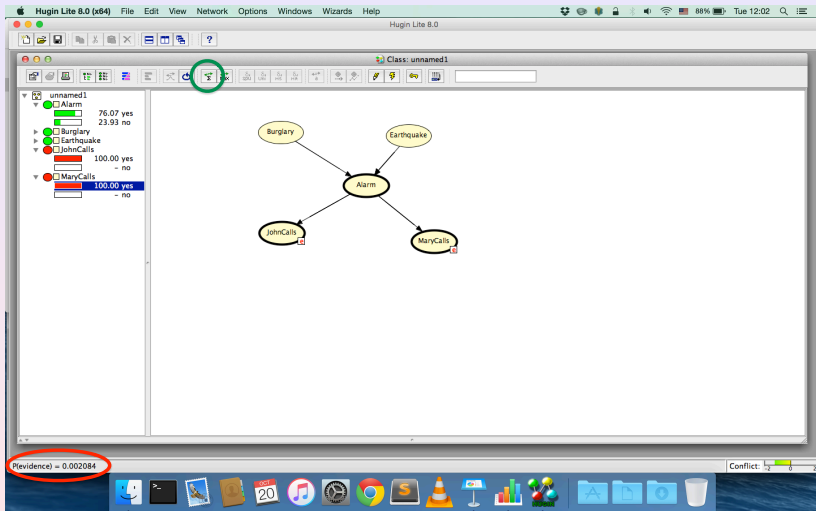
Running the Network

The screenshot displays the Hugin Lite 8.0 interface. The main window shows a Bayesian network with five nodes: Burglary, Earthquake, Alarm, JohnCalls, and MaryCalls. The nodes are arranged in a diamond shape with Alarm in the center. Arrows point from Burglary and Earthquake to Alarm, and from Alarm to JohnCalls and MaryCalls.

On the left side, a table lists the variables and their conditional probabilities:

Variable	Yes	No
Alarm	0.25	99.75
Burglary	0.10	99.90
Earthquake	0.20	99.80
JohnCalls	5.21	94.79
MaryCalls	1.17	98.83

P(evidence)



Computing the probability of a combination of states

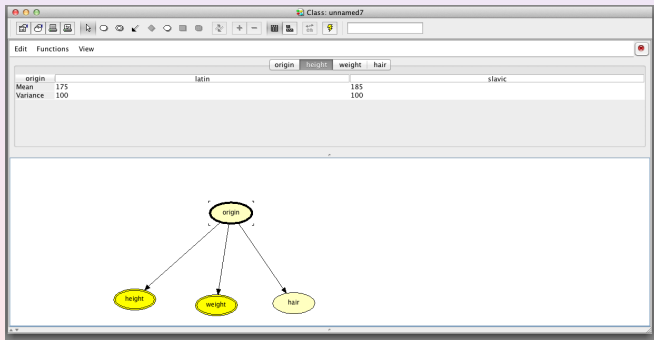
- We want to compute $P(\text{alarm} = \text{"yes"}, \text{johncalls} = \text{"yes"} | \text{burglary} = \text{"yes"})$
- Exploiting that $P(A, B) = P(A|B)P(B)$

$$\begin{aligned} P(\text{alarm} = \text{"yes"}, \text{johncalls} = \text{"yes"} | \text{burglary} = \text{"yes"}) &= \\ &= \frac{P(\text{alarm} = \text{"yes"}, \text{johncalls} = \text{"yes"}, \text{burglary} = \text{"yes"})}{P(\text{burglary} = \text{"yes"})} \end{aligned}$$

$$\begin{aligned} P(\text{alarm} = \text{"yes"}, \text{johncalls} = \text{"yes"} | \text{burglary} = \text{"yes"}) &= \\ &= \frac{0.000846}{0.001} = 0.846 \end{aligned}$$

Hybrid Networks

- Continuous nodes with mean and variance (Gaussian distributions)
- Continuous nodes can be children of discrete ones, not viceversa



Learning Wizard

- 1 Select Wizards, Learning Wizard
- 2 Load the training file (`small_asia.dat`)
- 3 In structure constraints import model information (from `ChestClinic.net`)
- 4 Select a learning algorithm
- 5 RUN the learning algorithm
- 6 Compile the learned network

Warning

- Without priors, some configurations get zero probability
- Add priors (experience) before running the learning (e.g. prior of 1 to each configuration)

Analysis Wizard

- 1 Select Wizards, Analysis Wizard
- 2 Sample 100 new examples according to the learned network
- 3 Check them in Data Source
- 4 Analyze the quality of the generated data in Data Accuracy
- 5 Clear the Data Source and Load the test file
(`test_asia_small.dat`)
- 6 Analyze the performance of classification of the learned network

Exercise

- 1 Consider the data file `leukemia.dat`
- 2 Each example contains 5 genes (active/inactive) and a label (AML/ALL)
- 3 **Randomly split** the file in train and test (80% train, 20% test)
- 4 Learn Bayesian network on train with different learning algorithms:
 - NPC
 - Greedy search-and-score
 - Fixed Naive Bayes structure (**NOTE: this is NOT tree-augmented Naive Bayes, see slides on Naive Bayes**)
- 5 Test the learned Bayesian networks on test