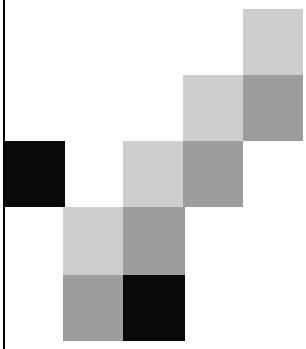


BNJ 2.03a Beginner Developer Tutorial

Chris H. Meyer
(revised by William H. Hsu)
Kansas State University
KDD Laboratory
http://www.kddresearch.org
http://bndev.sourceforge.net

Contents

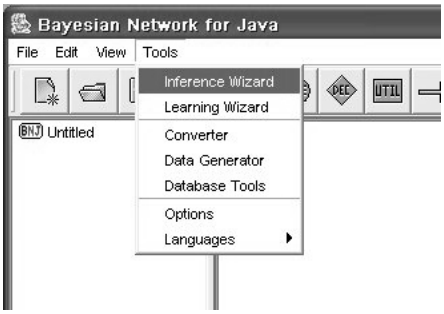
BNJ 2.0a Tools

- Offers many new tools in GUI form
- This lecture will focus on the Inference and Learning Wizards
- We will also look at components such as evidence, CPT tables, and algorithms behind learning

Contents

Starting the Inference Wizard (1)

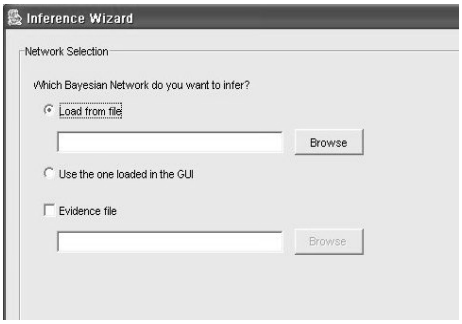
- Select
  Tools →
  Inference Wizard



http://bndev.sourceforge.net

Starting the Inference Wizard (2)

- **Load**
  existing network
  or
  GUI network

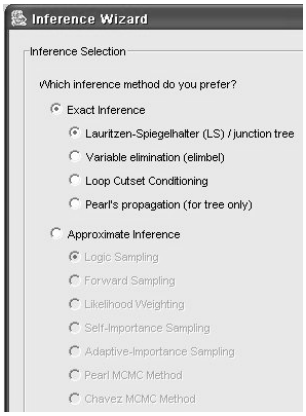


- You may also select to have an evidence file present

http://bndev.sourceforge.net

Using the Inference Wizard (1)

- Exact Inference Methods
  - LS / Junction Tree
  - Variable Elimination (elimbel)
  - Loop Cutset Conditioning
  - Pearl's Propagation (tree only)



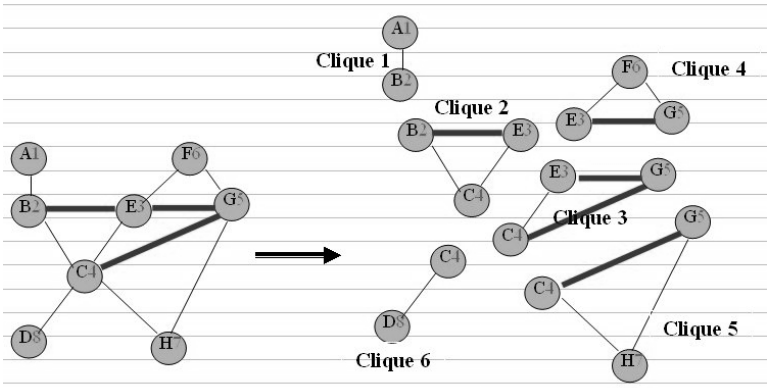
http://bndev.sourceforge.net

Using the Inference Wizard (2)

- L-S Algorithm contains 2 main steps:
  - Creates a tree of cliques (junction tree) from the Bayesian Network
  - Computes probability of cliques, then single-node properties are formed based on probability of cliques

http://bndev.sourceforge.net

Using the Inference Wizard (3)



(Example of Cliques in L-S algorithm)

Courtesy of Haipeng Guo

http://bndev.sourceforge.net

Using the Inference Wizard (4)

- Variable Elimination
  - Uses confactors and the VE algorithm instead of trees
- Loop Cutset
  - Finds minimum cutsets of probability in Bayesian Networks and computes probability from the cutsets

http://bndev.sourceforge.net

Using the Inference Wizard (5)

- Pearl's Propagation
  - Uses message-passing as data from 1 vertex propagates to all neighbors, then to neighbor's neighbors, etc…
- All algorithms are useful in the correct circumstance, but full explanation is not in the scope of this lecture

Using the Inference Wizard (6)

- Approximate Inference
  - Logic Sampling, Forward Sampling, Likelihood Weighting, Self-Importance Sampling, Adaptive-Importance Sampling, Pearl MCMC Method, Chavez MCMC Method
- Again, all algorithms are useful in the correct circumstance
- Output your data to chosen file on completion

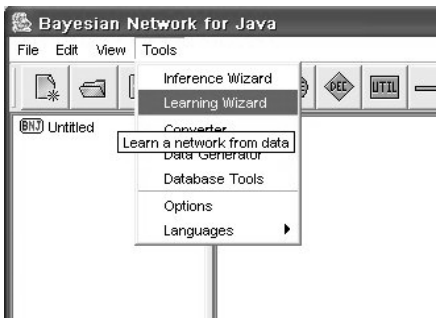
Contents

- Introduction
- Inference Tutorial
- **Learning Tutorial**
- Coding the Wizards

http://bndev.sourceforge.net

Starting the Learning Wizard (1)

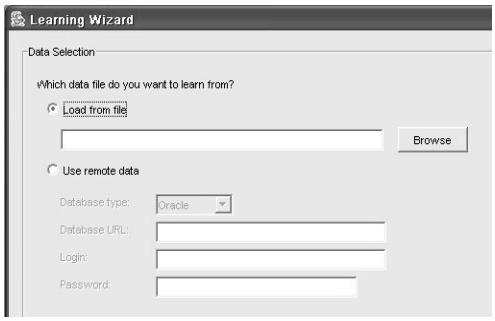
- Select
  Tools →
  Learning Wizard



http://bndev.sourceforge.net

Starting the Learning Wizard (2)

- Load
  Local File
  Or
  Database File



http://bndev.sourceforge.net

Using the Learning Wizard (1)

- Select your Learning Algorithm
  - K2, Genetic Algorithm Wrapper for K2 (GAWK), Genetic Algorithm on Structure, Greedy Structure Learning, Standard Hill-Climbing, Hill-Climbing with adversarial reweighting, Hill-Climbing with Dirichlet prior, Simulated Annealing, Stochastic structural learning

http://bndev.sourceforge.net

Using the Learning Wizard (2)

- Depending on user's desire, different learning algorithms will prove to be more effective
- Output results to file with desired ordering

Contents

Inference Wizard Coding (1)

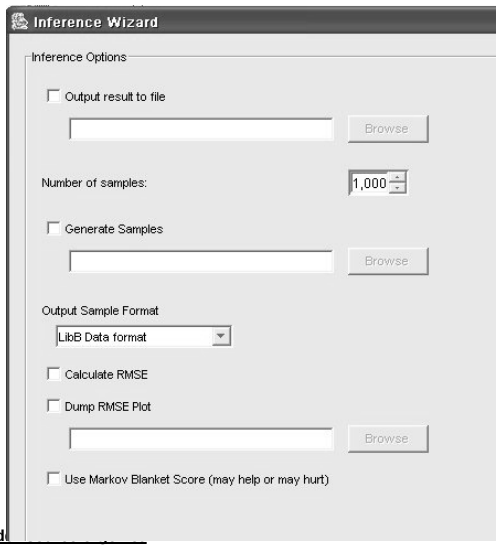
- GUI
  - Main GUI window passed as owner of InferenceWizard.java
  - All Buttons, JButtons, JRadioButtons, etc. are added to InferenceWizard.java's ActionListener(actionEvent e) method

Inference Wizard Coding (2)

- Secondary Windows
  - Built using BNJFileDialogFactory class
- GUI = First 500 lines of code for InferenceWizard.java
- Main Brain = Last 100 lines of code

Inference Wizard Coding (3)

- Approximate Inference is slightly more complicated →



http://bnd

Inference Wizard Coding (4)

- JTree, DefaultTreeModel, and InferenceResult are used for inference calculation
- These are found in javax.swing.tree.* and edu.ksu.cis.bnj.bbn.inference.*

http://bndev.sourceforge.net

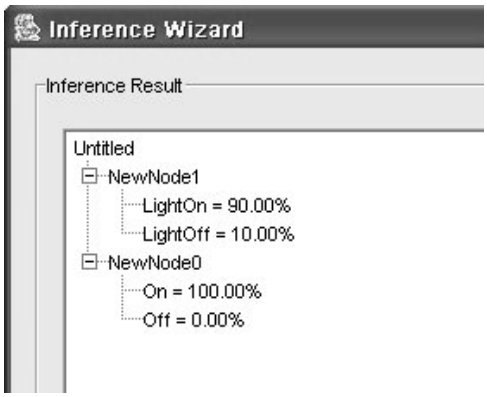
Inference Wizard Coding (5)

- Inference.getMarginals() returns an InferenceResult, which contains keys used for enumeration
- InferenceWizard parses all nodes, and adds them to a DefaultTreeModel.
- On completion of parsing, the JTree is updated with this model

http://bndev.sourceforge.net

Inference Wizard Coding (6)

- JPanel is created upon model completion for results



- Check edu.ksu.cis.bnj.bbn.inference.* for algorithm explanations

http://bndev.sourceforge.net

Coding the Learning Wizard (1)

- GUI
  - Almost exactly the same as InferenceWizard.java, except…
  - Database functionality has been implemented
    - Currently supports Oracle, MySQL, and PostgreSQL

Coding the Learning Wizard (2)

- Except for Database methods, connect() and disconnect(), the Learning Wizard is basically a mirror of the Inference Wizard
- GUI = First 400 lines of code for LearningWizard.java
- Main Brain = Last 30 lines of code

Coding the Learning Wizard (3)

- Key element of LearningWizard.java = Learner
  - Imported from edu.ksu.cis.bnj.bbn.learning.*
- Learner is then instantiated with the selected algorithm and data as parameters
  - i.e. :

```
learner = Learner.load(learningEngines[selectedAlgorithm][1], data);
                                    Or
learner = new PRMk2((Database) data, null);
```


Coding the Learning Wizard (4)

- Learned graph is returned via

  Learner.getGraph()

- This statement returns a BBNGraph suitable for viewing
- Check edu.ksu.cis.bnj.bbn.learning.* for algorithm explanations