COMP-526 Bayes Net in Java

Rohan Shiloh Shah
rshah3@cs.mcgill.ca
March 10, 2003

1 EXECUTING THE PROGRAM

```
froggy@LUCY$ javac *.java

froggy@LUCY$ java Menu
```

To follow the execution of the program issue the following command

```
froggy@LUCY$ java Menu -d
```

and all debugging information will be output to screen.
You can save and load Bayesian Networks to disk using the menu.[1]

2 DATA STRUCTURE

Entries in the Conditional Probability Table are arranged in a Hashtable where the factors are keys and are mapped to their probabilities. The factors or keys have an instantiated 'String Representation' of the form `wetgrass=false|rain=true&sprinkler=true`. In class BeliefNet, method getStringRepresentation(String nodeName) generates the uninstantiated 'String Representation' of a node in the form $var|Parents(var)$. The method getVectorRepresentation() returns the uninstantiated 'String Representation' of all the nodes in the network.

Each variable (modelled by class Node) has its own local CPT. In class BeliefNet there is a global CPT which contains all the local CPTs of all the variables. New factors can be added to the global CPT using method addFactorToGlobalCPT(String key, String prob). This storing of intermediate factors in a global table is the basis for the dynamic programming view of the variable elimination algorithm.

To look up a probability use the method findProbInGlobalCPT(String instRep) in class BeliefNet where instRep is a fully instantiated factor in its 'String Representation'. Order is irrelevant, ie instRep can be either `Smoke=true&Fire=false` or `Fire=false&Smoke=true`. findProbInGlobalCPT(String instRep) will generate all permutations and check the global CPT for each one.

If there are no parents then 2 entries exist in the CPT, `variable=true` and `variable=false`.

---

[1] If you change the Node or BeliefNet class then any Bayesian Networks that were previously saved might not load properly and new Bayesian Networks will have to be created.

3 Variable Elimination Algorithm

The following algorithms are detailed versions of algorithms found in [1] (Lecture Notes #3).

3.1 *Without Evidence*

Require: an un-instantiated query string Q in its String Representation, i.e. $Q_1 \& Q_2 \& \cdots \& Q_n$

```
if GlobalCPT contains Q then
  print probability associated with Q and exit
end if
$\mathbf{Y}$ is a list of the query variables in Q
$\mathbf{X}$ is a list of the variables/nodes in the Bayes Net
Initialize $\mathbf{F}$ with strings $X_i|Parents(X_i)$[2] $\forall i$
$\mathbf{Z} = \mathbf{X} - \mathbf{Y}$
for $i = 0$ to $|\mathbf{Z}|$ do
  Choose variable $Z_i$ for elimination
  Remove from $\mathbf{F}$ all strings mentioning $Z_i$ and put them in $\mathbf{F}'$
  Put all uninstantiated variables from $\mathbf{F}'$ not including $Z_i$ into $\mathbf{U}$
  Make two copies of $\mathbf{F}'$ and call them $\mathbf{F}''_{\mathbf{TRUE}}$ and $\mathbf{F}''_{\mathbf{FALSE}}$
  Assign all instances of $Z_i$ in $\mathbf{F}''_{\mathbf{TRUE}}$ the value TRUE
  Assign all instances of $Z_i$ in $\mathbf{F}''_{\mathbf{FALSE}}$ the value FALSE

  for $j = 0$ to $2^{|U|}$ do
    generate one (out of a total of $2^{|U|}$) instantiation[3] for the variables in $\mathbf{U}$: $I_j$
    We will call the instantiated String Representation of $I_j$: $\mathtt{SRI}(I_j)$
    We will call the un-instantiated String Representation of $I_j$: $\mathtt{SRU}(I_j)$
    Assign the instantiation $I_j$ to the appropriate variables in both $\mathbf{F}''_{\mathbf{TRUE}}$ and $\mathbf{F}''_{\mathbf{FALSE}}$
    $Prob(\mathtt{SRI}(I_j)) = \sum_{Z_i} \prod \mathbf{F}''_{\mathbf{Z_i}} = \prod \mathbf{F}''_{\mathbf{TRUE}} + \prod \mathbf{F}''_{\mathbf{FALSE}}$
    Add $Prob(\mathtt{SRI}(I_j))$ to the globalCPT with the $\mathtt{SRI}(I_j)$ as the key
    Add $\mathtt{SRU}(I_j)$ to $\mathbf{F}$
  end for
end for
Note: see method normaliseForNoEvidence()
Put all uninstantiated variables from $\mathbf{F}$ into $\mathbf{U}$
for $j = 0$ to $2^{|U|}$ do
  Make a copy of $\mathbf{F}$: $\mathbf{F}^{copy}$
  Generate one instantiation for the variables in $\mathbf{U}$: $I_j$
  Assign the instantiation $I_j$ to the appropriate variables in $\mathbf{F}^{copy}$
  Assign the instantiation $I_j$ to the variables in Q: $\mathtt{Q}_j$
  $Prob(\mathtt{Q}_j) = \prod_i F_i^{copy}$
end for
```

---

[2] this is the uninstantiated 'String Representation' (or factor) of a node/variable. As an example, for wetgrass it would be wetgrass|rain&sprinkler.

[3] Instantiations must be generated in a predetermined order, so that there are no repeated instantiations produced.

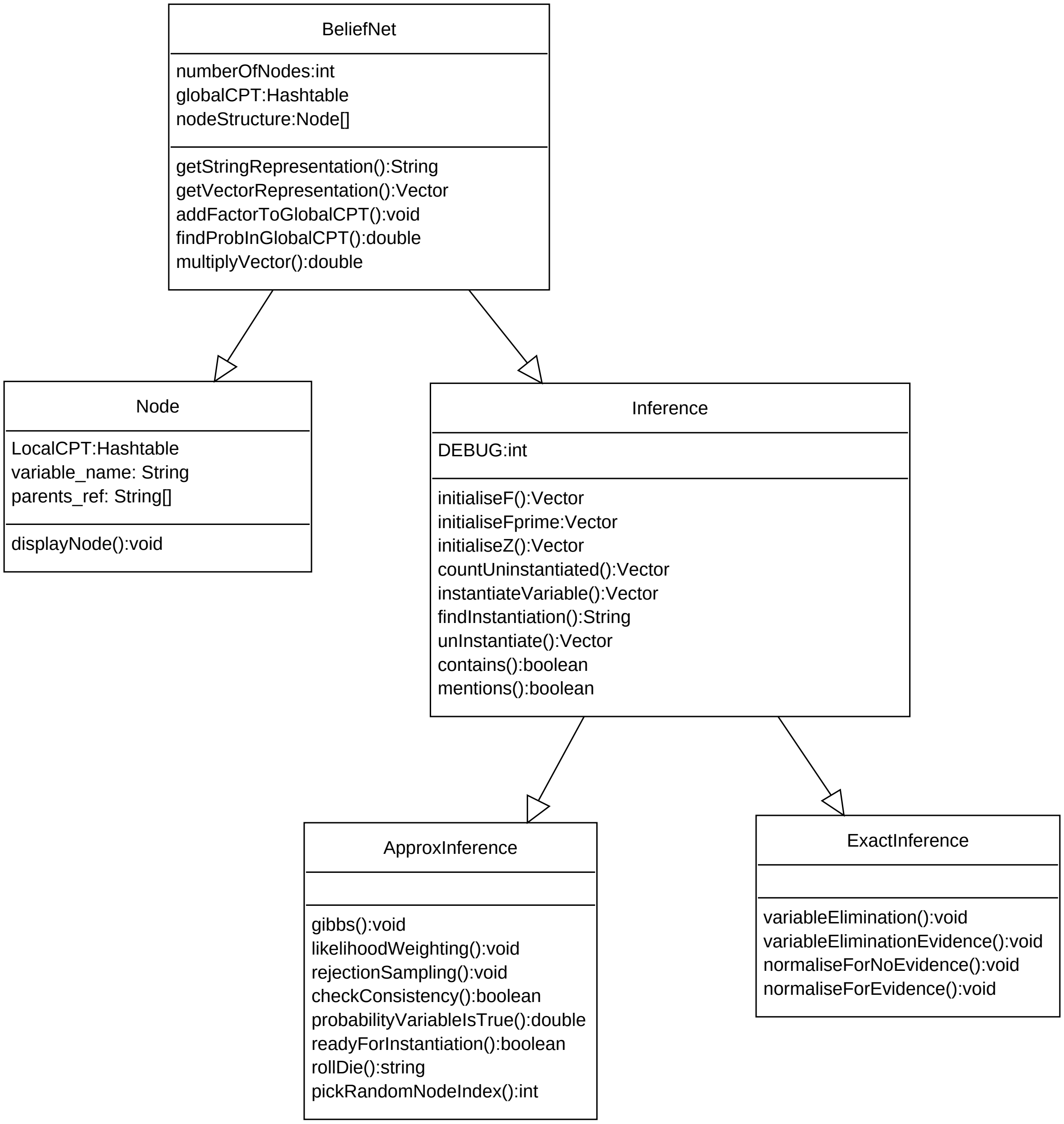

Figure 1: UML Diagram

3.2 *With Evidence*

Differences from the Variable Elimination algorithm without evidence are marked with a star ⋆

```
Require: an instantiated query Q and evidence string E both in their String Representations, i.e.
  $Q_1 = boolean\&Q_2 = boolean\& \cdots \&Q_n = boolean$

  if ⋆ GlobalCPT contains Q|E then
    print probability associated with Q and exit
  end if
  ⋆ $\mathbf{Y}$ is a list of the query variables in Q and evidence variables in E
  $\mathbf{X}$ is a list of the variables/nodes in the Bayes Net
  Initialize $\mathbf{F}$ with strings $X_i|Parents(X_i)\forall i$
  ⋆ Instantiate evidence variables in $\mathbf{F}$ with the instantiations found in E[4]
  $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$
  for $i = 0$ to $|\mathbf{Z}|$ do
    Choose variable $Z_i$ for elimination
    Remove from $\mathbf{F}$ all strings mentioning $Z_i$ and put them in $\mathbf{F}'$
    ⋆ $\mathbf{G_I}$[5] is an instantiated String Representation of the variables that are common to both E and $\mathbf{F}'$
    Put all uninstantiated variables from $\mathbf{F}'$ not including $Z_i$ into $\mathbf{U}$
    Make two copies of $\mathbf{F}'$ and call them $\mathbf{F}''_{\mathbf{TRUE}}$ and $\mathbf{F}''_{\mathbf{FALSE}}$
    Assign all instances of $Z_i$ in $\mathbf{F}''_{\mathbf{TRUE}}$ the value TRUE
    Assign all instances of $Z_i$ in $\mathbf{F}''_{\mathbf{FALSE}}$ the value FALSE
    for $j = 0$ to $2^{|U|}$ do
      generate one (out of a total of $2^{|U|}$) instantiation for the variables in $\mathbf{U}$: $I_j$
      We will call the instantiated String Representation of $I_j$: SRI($I_j$)
      We will call the un-instantiated String Representation of $I_j$: SRU($I_j$)
      Assign the instantiation $I_j$ to the appropriate variables in both $\mathbf{F}''_{\mathbf{TRUE}}$ and $\mathbf{F}''_{\mathbf{FALSE}}$
      ⋆ $Prob(\texttt{SRI}(I_j)\&\mathbf{G_I}) = \sum_{Z_i} \prod \mathbf{F}''_{\mathbf{Z_i}} = \prod \mathbf{F}''_{\mathbf{TRUE}} + \prod \mathbf{F}''_{\mathbf{FALSE}}$
      ⋆ Add $Prob(\texttt{SRI}(I_j)\&\mathbf{G_I})$ to the globalCPT with the SRI($I_j$)&$\mathbf{G_I}$ as the key
      ⋆ Add SRU($I_j$)&$\mathbf{G_I}$ to $\mathbf{F}$
    end for
  end for
  Note: see method normaliseForEvidence()
  ⋆ initialize totalsum to 0.0
  Put all uninstantiated variables from $\mathbf{F}$ into $\mathbf{U}$
  for $j = 0$ to $2^{|U|}$ do
    Make a copy of $\mathbf{F}$: $\mathbf{F}^{copy}$
    Generate one instantiation for the variables in $\mathbf{U}$: $I_j$
    Assign the instantiation $I_j$ to the appropriate variables in $\mathbf{F}^{copy}$
    Assign the instantiation $I_j$ to the variables in Q: $Q_j$
    ⋆ $Prob(Q_j\&E) = \prod_i F_i^{copy}$ [6]
    ⋆ totalsum $+=$ $Prob(Q_j\&E)$
  end for
  ⋆ $Prob(E) =$ totalsum
  ⋆ $Prob(Q_j|E) = Prob(Q_j\&E)/Prob(E)$
```

---

[4] For example, if `E=var1=true&var2=false then instantiate var1 and var2 to give F=[var1=true|var3, var3&var4&var2=false]`: **F** is now partially instantiated.

[5] We have not accounted for the evidence variables that are in $\mathbf{F}'$ since they are instantiated and therefore will not be included in **U**. So for example if `E=wetgrass=true&cloudy=false&rain=true` and $\mathbf{F}'$`=[sprinkler=true|cloudy=false, wetgrass=true|sprinkler=true]` then $\mathbf{G_I}$`=wetgrass=true&cloudy=false`

[6] The $Prob(Q_j\&E)$ values should be stored so they can be used later; try using a Hashtable where the key is the String Representation of Q_j&E

4 APPROXIMATE INFERENCE

The following algorithms are detailed versions of algorithms found in [1] (Reading # 7).

4.1 *Rejection Sampling*

```
Require: an instantiated query Q and evidence string E both in their String Representations, i.e. in
  the form $Q_1 = boolean\&Q_2 = boolean\& \cdots \&Q_n = boolean$
Require: the number of iterations to evalute; numberOfIterations

  Z is a list of the variables/nodes in the Bayes Net
  validSamples counts the number of instances that are consistent with the evidence
  consistentWithQuery counts the number of instances that are consistent with both the evidence and
  the query

  for $i = 0$ to numberOfIterations do
    Initialize F with strings $Z_i|Parents(Z_i)\forall i$
    Initialize a Queue Q with all the variables in Z

    while Q is not empty do
      if Q.head ready for instantiation[7] then
        Find Prob(Q.head=TRUE)
        if rollDie < Prob(Q.head=TRUE) then
          instantiate Q.head with TRUE in F
        else
          instantiate Q.head with FALSE in F
        end if
        Dequeue Q.head
      else
        Dequeue Q.head and add to the end of queue
      end if
    end while

    if F is consistent with evidence then
      validSamples++
      if F is consistent with query then
        consistentWithQuery++
      end if
    end if
  end for
  Prob(Q|E)=consistentWithQuery/validSamples
```

---

[7] A variable is ready for instantiation when all its parents have been instantiated. Check for instantiations in **F**

4.2 *Likelihood Weighting*

Differences from the Rejection Sampling algorithm are marked with a star ⋆

```
Require: an instantiated query Q and evidence string E both in their String Representations, i.e. in
   the form $Q_1 = boolean\&Q_2 = boolean\& \cdots \&Q_n = boolean$
Require: the number of iterations to evalute; numberOfIterations

   Z is a list of the variables/nodes in the Bayes Net
   ⋆ totalSumOfW counts the total weight w
   ⋆ consistentWithQuery counts the weight w of instances that are consistent with both the evidence
   and the query [8]

   for $i = 0$ to numberOfIterations do
      ⋆ double w = 1.0
      Initialize F with strings $Z_i|Parents(Z_i)\forall i$
      Initialize a Queue Q with all the variables in Z

      while Q is not empty do
         if Q.head ready for instantiation then
            Find Prob(Q.head=TRUE)
            if ⋆ E mentions Q.head=TRUE then
               ⋆ w = w · Prob(Q.head=TRUE)
            else if ⋆ E mentions Q.head=FALSE then
               ⋆ w = w · (1−Prob(Q.head=TRUE))
            else
               if rollDie < Prob(Q.head=TRUE) then
                  instantiate Q.head with TRUE in F
               else
                  instantiate Q.head with FALSE in F
               end if
               Dequeue Q.head
            end if
         else
            Dequeue Q.head and add to the end of queue
         end if
      end while

      if F is consistent with query then
         ⋆ consistentWithQuery + = w
      end if
      ⋆ totalSumOfW + = w
   end for
   ⋆ Prob(Q|E)=consistentWithQuery/totalSumOfW
```

---

[8] Actually all instances are consistent with the evidence.

4.3 *Gibbs Sampling*

```
Require: an instantiated query Q and evidence string E both in their String Representations, i.e. in
  the form $Q_1 = boolean\&Q_2 = boolean\& \cdots \&Q_n = boolean$
Require: the number of iterations to evalute; numberOfIterations

  Z is a list of the variables/nodes in the Bayes Net
  C = Z - E[9]
  Initialize consistentWithQuery= 0.0
  Initialize F with strings $Z_i|Parents(Z_i) \forall i$
  Instantiate the evidence variables in F

  Initialize a Queue Q with all the variables in C
  while Q is not empty[10] do
    if Q.head ready for instantiation then
      Find Prob(Q.head=TRUE)
      if rollDie < Prob(Q.head=TRUE) then
        instantiate Q.head with TRUE in F
      else
        instantiate Q.head with FALSE in F
      end if
      Dequeue Q.head
    else
      Dequeue Q.head and add to the end of queue
    end if
  end while
  for $i = 0$ to numberOfIterations[11] do
    Choose a random non-evidence variable: $\mathbf{C}_i$
    Un-instantiate $\mathbf{C}_i$ in F
    Find Prob($\mathbf{C}_i$=TRUE)
    if rollDie < Prob($\mathbf{C}_i$=TRUE) then
      instantiate $\mathbf{C}_i$ with TRUE in F
    else
      instantiate $\mathbf{C}_i$ with FALSE in F
    end if

    if F is consistent with query then
      consistentWithQuery ++
    end if
  end for
  Prob(Q|E)=consistentWithQuery/numberOfIterations
```

---

[9] **E** are the variables in the evidence string E

[10] We will choose a starting point in our random walk by randomly instantiating all the non-evidence variables. This is one way to choose the starting point; another possibility would be to simply include the instantiations that are present in Q.

[11] The next node on our random walk is generated by randomly choosing a non-evidence variable (from **C**) and randomly sampling its instantiation.

5 CREATING A NEW BAYES NET

The variable names should be atleast 4 characters long and not substrings of either 'true' or 'false'. This is simply a precaution to avoid the variable names getting mixed up with the predefined strings =TRUE and =FALSE.

```
Create a NEW Belief-Net? (1)
LOAD a Belief-Net from disk? (2) 1
How many variables (or nodes) does this BayesNet have ?4

Node 1
What is the name of the new variable(or Node)? cloudy
How many parents does this variable have? 0
How many children does this variable have? 2
Enter child name 1 ? rain
Enter child name 2 ? sprinkler
Please enter the following probabilities -

P(cloudy=true) = ? 0.32

Node 2
What is the name of the new variable(or Node)? rain
How many parents does this variable have? 1
Enter parent name 1 ? cloudy
How many children does this variable have? 1
Enter child name 1 ? wetgrass
Please enter the following probabilities -

P(rain=true|cloudy=false) = ? 0.37
P(rain=true|cloudy=true) = ? 0.45

Node 3
What is the name of the new variable(or Node)? sprinkler
How many parents does this variable have? 1
Enter parent name 1 ? cloudy
How many children does this variable have? 1
Enter child name 1 ? wetgrass
Please enter the following probabilities -

P(sprinkler=true|cloudy=false) = ? 0.32
P(sprinkler=true|cloudy=true) = ? 0.66

Node 4
What is the name of the new variable(or Node)? wetgrass
How many parents does this variable have? 2
Enter parent name 1 ? rain
Enter parent name 2 ? sprinkler
How many children does this variable have? 0
Please enter the following probabilities -

P(wetgrass=true|rain=false&sprinkler=false) = ? 0.34
P(wetgrass=true|rain=false&sprinkler=true) = ? 0.01
```

```
P(wetgrass=true|rain=true&sprinkler=false) = ? 0.98
P(wetgrass=true|rain=true&sprinkler=true) = ? 0.67
Would you like to save the Belief-Net to disk? (y/n) y
Enter a filename?net

Make a selection:
(1) Exact Inference
(2) Approximate Inference
(3) Quit
```

References

[1] Daphne Koller, Nir Friedman - Lecture Notes, 2000

[2] Finn V. Jensen - Bayesian Networks and Decision Graphs, 2001

[3] Doina Precup - Lecture Notes, 2001