Assignment 3: Bayes Nets

CSC 384H—Winter 2015

Out: March 16th, 2015
Due: Electronic Submission Wednesday April 1st, 7:00pm
Late assignments will not be accepted without medical excuse
Worth 15% of your final.

Be sure to include your name and student number as a comment in all submitted documents.

Handing in this Assignment

What to hand in on paper: No paper submission required.

What to hand in electronically: You must submit all your answers and code electronically. You must submit the following:

1. A file of python code called **`bnetbase.py`**. This file will contain your implementation of the factor functions and variable elimination.
2. A PDF file called **`a3_answers.pdf`** Use any word processing software to produce your written answers and then convert to PDF.

To submit these files electronically, use the CDF secure Web site
`https://www.cdf.toronto.edu/students/`
or use the CDF **submit** command. Type **man submit** for more information. The name of the assignment for submit will be "A3"

Since we will test your code electronically, you must

- *make certain that your code runs on CDF using python3 (version 3.4.1* (installed as "python3" on CDF, note that "python" with no "3" invokes the wrong version of python).
- not add any non-standard pythons imports from within the python files you submit (the imports that are already in the template files must remain).
- include all your written answers in the PDF file you create and submit (using the file name `a3_answers.pdf`). **NOTE the space limits specified for each.** If you exceed the space limits marks will be deducted.

Introduction

In this assignment you will implement variable elimination on Bayes Nets.

What is supplied. You will be supplied with python code implementing **Variable**, **Factor**, and **BN** objects. The file `bnetbase.py` contains the class definitions for these objects. The code supports representing factors as tables of values indexed by various settings of the variables in the factor's scope.

The template file `bnetbase.py` also contains function prototypes for the functions you must implement.

Question 1. Implement Variable Elimination

Implement a collection of functions that operate on `Factor` objects and then use these functions to implement `VE` (variable elimination).

`multiply_factors` Take as input a list of `Factor` objects; create and return a new factor that is equal to the product of the factors in the list. Do not modify any of the input factors.

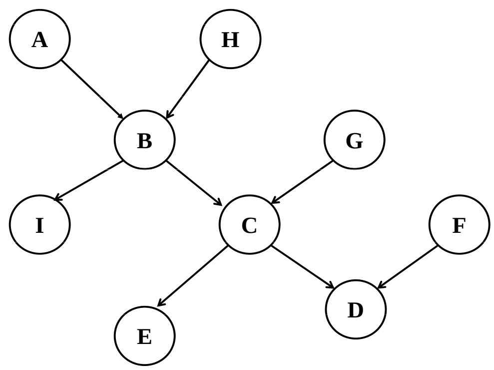
`restrict_factor` Take as input a single factor, a variable V and a value d from the domain of that variable; create and return a new factor that is the restriction of the input factor to the assignment $V = d$. Do not modify the input factor.

sum_out_variable Take as input a single factor, and a variable V; create and return a new factor that is the result of summing V out of the input factor. Do not modify the input factor.

VE Take as input a Bayes Net object (object of class `BN`), a variable that is the query variable Q, and a list of variables E that are the evidence variables (all of which have had some value set as evidence using the variable's `set_evidence` interface). Compute the probability of every possible assignment to Q given the evidence specified by the evidence settings of the evidence variables. Return these probabilities as a list where every number corresponds the probability of one of Q's possible values. Do not modify any factor of the input bayes net.

Simple Test Bayes Net

Use your implementation to answer questions regarding the following Bayesian network. All variables are binary, with values $Dom(A) = \{a, \bar{a}\}, Dom(B) = \{b, \bar{b}\}$, etc.



The probability table values are as follows (probabilities add to one so you can compute the remaining probability values):

$$Pr(a) = 0.9$$
$$Pr(b|ah) = 1.0, Pr(b|a\bar{h}) = 0.0, Pr(b|\bar{a}, h) = 0.5, Pr(b|\bar{a}\bar{h}) = 0.6$$
$$Pr(c|bg) = 0.9, Pr(c|b\bar{g}) = 0.9, Pr(c|\bar{b}, g) = 0.1, Pr(c|\bar{b}\bar{g}) = 1.0$$
$$Pr(d|cf) = 0.0, Pr(d|c\bar{f}) = 1.0, Pr(d|\bar{c}, f) = 0.7, Pr(d|\bar{c}\bar{f}) = 0.2$$
$$Pr(e|c) = 0.2, Pr(e|\bar{c}) = 0.4$$
$$Pr(f) = 0.1$$
$$Pr(g) = 1.0$$
$$Pr(h) = 0.5$$
$$Pr(i|b) = 0.3, Pr(i|\bar{b}) = 0.9$$

1. Using your Variable Elimination implementation, compute the following probabilities:
   (a) $Pr(b|a)$
   (b) $Pr(c|a)$
   (c) $Pr(c|a\bar{e})$
   (d) $Pr(c|a\bar{f})$

To Submit

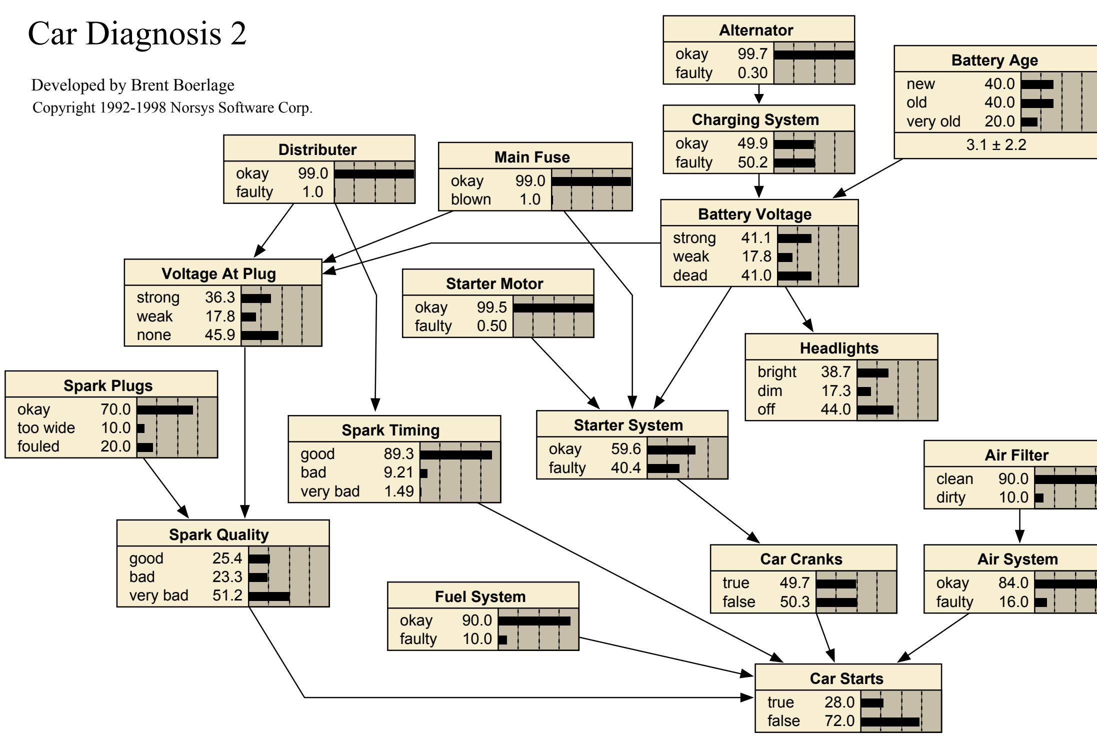
1. Submit your python implementation in the file `bnetbase.py`, along with the rest of your implementation.

2. Submit your answers for the simple test Bayes Net in your write up file `a3_answers.pdf`.

Question 2. Car Diagnosis Network

In the file `carDiagnosis.py` you will find a specification of a Bayes Net for diagnosing various reasons why a car might not start.

The layout of this Bayes Net is shown below:



In this diagram each variable of the Net is shown in a square box along with the values that the variable can take. For example, The variable `Battery Voltage` (which is the variable `bv` in the file `carDiagnosis.py`) can take on one of three different values: “strong”, “weak” and “dead”.

The numbers and bars show the *unconditional* probabilities of the variables taking on their different values. For example, in the Net we have $P(\text{Battery Voltage} = \text{dead}) = 41.0\%$.

Given this Net you are to come up with some test cases. Each test case is to illustrate a particular feature of the network as described below. For each test case you are to use your implementation of `VE` to compute probabilities that illustrate the feature your test case is designed to illustrate.

For example, say you were asked give a test case illustrating that in the network some variables are independent of each other. We know that in a Bayes net a variable is conditionally independent of its *non-*descendants given values for its parents (slide 148 of the lecture notes). Thus any two root nodes in the network, i.e., nodes with no parents, should be independent of each other (i.e., *unconditionally* independent since there are no parents to condition on). So one correct answer to the question would be the following (ds = Distributor and al = Alternator):

$$P(ds = \text{okay}) = P(ds = \text{okay}|al = \text{okay}) = P(ds = \text{okay}|al = \text{faulty}) = 0.99$$
$$P(ds = \text{faulty}) = P(ds = \text{faulty}|al = \text{okay}) = P(ds = \text{faulty}|al = \text{faulty}) = 0.01$$

So ds is independent of al.

In this answer the implementation has been used to compute all of the probabilities. These probabilities turned out to satisfy these above equations thus demonstrating that the Net captures this independence.

Give a test case and an answer like the above for the following four things you are to illustrate about the Net. *No extra statements should be made about your test cases; just show the relations between the probabilities that hold in the Net along with up to two sentences to state what these relations illustrate.*

Format your answers clearly or you will lose marks.

1. Show a case of conditional independence in the Net where knowing some evidence item $V_1 = d_1$ makes another evidence item $V_2 = d_2$ irrelevant to the probability of some third variable V_3. (Note that conditional independence requires that the independence hold for all values of V_3).

2. Show a case of *explaining away*, where knowing an evidence item $V = d$ increases the probability of its possible causes $V_1 = d_1, \ldots, V_k = d_k$ but subsequently increasing the probability of one of these clauses $V_i = d_i$ decreases the probability of the other causes as it explains away the evidence $V = d$.

3. Show a sequence of accumulated evidence items $V_1 = d_1, \ldots, V_k = d_k$ (i.e., each evidence item in the sequence is added to the previous evidence items) such that each additional evidence item increases the probability of some variable V having value d. (That is, the probability of $V = d$ increases monotonically as we add the evidence items).

4. Show a sequence of accumulated evidence items $V_1 = d_1, \ldots, V_k = d_k$ such that as we add the evidence items the probability of some variable V having value d both increases and decreases. (That is, some of the evidence makes $V = d$ more probable while other evidence items make $V = d$ less probable).

Submit the your answers to the four test case questions in your written answers file `a3_answers.pdf`.

Soon Chee Loong c4soonch
999295793 CSC384 Winter 2015

Assignment 3: Bayes Net

CSC384 Winter 2015 Assignment 3 Question 1

a) Pr(b|a)

0.5

b) Pr(c|a)

0.5

c) Pr(c|a,-e)

0.5714285714285714

d) Pr(c|a,-f)

0.5

CSC384 Winter 2015 Assignment 3 Question 2

1.

V1 = ss , V2 = sm , V3 = cc

P(cc|ss) = P(cc|ss,sm)

P(cc = true|ss = okay) = 0.80 = P(cc = true|ss = okay, sm = okay) = 0.80

P(cc = false|ss = okay) = 0.20 = P(cc = false|ss = okay, sm = okay) = 0.20

P(cc = true|ss = okay) = 0.80 = P(cc = true|ss = okay, sm = faulty) = 0.80

P(cc = false|ss = okay) = 0.20 = P(cc = false|ss = okay, sm = faulty) = 0.20

P(cc = true|ss = faulty) = 0.05 = P(cc = true|ss = faulty, sm = okay) = 0.05

P(cc = false|ss = faulty) = 0.95 = P(cc = false|ss = faulty, sm = okay) = 0.95

P(cc = true|ss = faulty) = 0.05 = P(cc = true|ss = faulty, sm = faulty) = 0.05

P(cc = false|ss = faulty) = 0.95 = P(cc = false|ss = faulty, sm = faulty) = 0.95

These statements illustrate that probability of *cc* is conditionally independent on *sm* given *ss*.

2.

V = sq , d = bad

V1 = pv , d1 = weak

V2 = sp , d2 = fouled

P(pv = weak) = 0.178

P(sp = fouled) = 0.200

P(pv = weak | sq = bad) = 0.533

P(sp = fouled | sq = bad) = 0.311

P(pv = weak | sq = bad , sp = fouled) = 0.000

P(sp = fouled | sq = bad, pv = weak) = 0.000

These statements illustrate that *sq = bad* increases the probability of *pv = weak* and *sp = fouled*. However, once it is known that *sp = fouled*, which explains away the reason *sq = bad*, probability of *pv = weak* decreases and vice-versa.

3.

V = ss , d = okay

V1 = sm , d1 = okay

V2 = mf , d2 = okay

V3 = bv , d3 = strong

P(ss = okay|sm = okay) < P(ss = okay|sm = okay, mf = okay) < P(ss = okay|sm = okay, mf = okay, bv = strong)

P(ss = okay|sm = okay) = 0.598

P(ss = okay|sm = okay, mf = okay) = 0.605

P(ss = okay|sm = okay, mf = okay, bv = strong) = 0.980

These statements illustrate that probability of *ss = okay* increases monotonically as we add the evidence items of *sm = okay*, *mf = okay*, and *bv = strong*.

4.

Soon Chee Loong c4soonch
999295793 CSC384 Winter 2015

V = st , d = true

V1 = sq , d1 = bad

V2 = fs , d2 = faulty

V3 = tm , d3 = good

V4 = asys , d4 = faulty

V5 = cc , d5 = true

P(st = true|sq = bad) > P(st = true|sq = bad, fs = faulty) < P(st = true|sq = bad, fs = faulty, tm = good) > P(st = true|sq = bad, fs = faulty, tm = good, asys = faulty) < P(st = true|sq = bad, fs = faulty, tm = good, asys = faulty, cc = true)

[0.2803836125216086, 0.7196163874783914]

P(st = true|sq = bad) = 0.462

P(st = true|sq = bad, fs = faulty) = 0.032

P(st = true|sq = bad, fs = faulty, tm = good) = 0.034

P(st = true|sq = bad, fs = faulty, tm = good, asys = faulty) = 0.015

P(st = true|sq = bad, fs = faulty, tm = good, asys = faulty, cc = true) = 0.020

These statements illustrate that probability of *st = true* both increases and decreases as we add the evidence items of *sq = bad*, *fs = faulty*, *tm = good*, *asys = faulty*, and *cc = true*.