CPSC 522 — Spring 2013

Assignment 1

Solution

Question 1

Suppose we want to diagnose school student's addition of multi-digit binary numbers. Suppose we are only considering subtracting a two-digit number from a three digit number.

That is, problems of the form:

$$\begin{array}{ccc} X_2 & X_1 & X_0 \\ - & Y_1 & Y_0 \\ \hline Z_2 & Z_1 & Z_0 \end{array}$$

Here X_i, Y_i and Z_i are all binary digits.

In this question you will represent this problem as a belief network, and test it with a belief network implementation.

(a) Describe how to do multi-digit binary subtraction. [This procedure that you are assuming students are carrying out will affect the network produced.] What errors would you expect students to make?

Solution: The student starts at the rightmost digit (digit 0). If the digit on the bottom is bigger, it needs to borrow from the left; i.e., set the borrow out to 1 (otherwise it is zero). Then the Z_0 value is the same as X_0 if Y_0 is 0 and is the opposite if Y_0 is 1.

For subsequent digits it works the same, except that it also takes into account whether the previous digit needed to borrow (i.e., the borrow out for the previous digit becomes the borrow-in for this digit). If the borrow-in was 0, it works the same as digit 0. If the borrow-in was 1, it needs to add one to the bottom digit; if this results in a 2, the set the borrow 1 and the Z_1 is the same as Y_1, and otherwise proceed as for digit 0.

I would expect students to make mistakes on basic subtraction (if they didn't even know how to subtract single digits), not borrowing when needed, and not taking into account the borrowed digit from the previous computation.

(b) What variables are needed to model subtraction of this form and the errors students could make? Give a DAG that specifies the dependence of these variables. [Hint: think about what will be observed, what will be queried, and hidden variables that you may have used in your description.]

Solution: See: `http://cs.ubc.ca/~poole/cs522/2013/as1/subtraction.xml` for this and the following two questions. You also need to test it to get full marks.

(c) What are reasonable conditional probabilities for this domain?

(d) Implement this, for example using the belief network tool at:
`http://www.aispace.org/bayes/` Test the implementation (don't forget to save your graph).

(e) Explain how to extend your graph to allow for two different subtraction problems to help us better test a student. [Hint: What nodes should be shared and what should be replicated?] It is up to you whether you implement this, but it's probably instructive to do so.

Solution: The X_i, Y_i and Z_i nodes should be duplicated, and the other nodes (pertaining to the student not the problem) should not.

Question 2

In this question we will explore ideas to extend the idea of belief networks. Be brief.

(a) What would happen if there were two distinct methods that students used to solve such subtraction problems and we are not certain which method a student will use? Explain how your network could be adapted to handle this.

Solution: I would have a new node *Method* which represents the solution method. The errors that the students make will be dependent on the solution method. For each solution method, I would create nodes for the skills the student requires when using that solution method. When these nodes are used, the solution method should also be used. E.g., if skill S is used for method 1, then for any node that relies on skill S, the *Method* node should also be a parent, and when $Method = 1$, the node should depend on S. [Interestingly, *Method* does not need to be a parent of S.] By observing multiple problems, we can query the posterior distribution on the method used, as well as the probabilities of the errors.

(b) What happens if we allow for a different base (say base 10 instead of base 2)? What is the size of the resulting tables? Suggest what could be done to make this more manageable.

Solution: $P(Z_1|X_1, Y_1, Borrow_1, Knows\_to\_borrow, Knows_subtraction)$ will be the largest table of size $2^3 10^3 = 8000$. There is much structure here; it would be good to use arithmetic to compute the correct answer and then, e.g., have a uniform distribution over mistakes. We could write a little program to compute the probability very easily.

(c) What if we were to allow for an arbitrary number of digits for each subtraction problem and an arbitrary number of different subtraction problems to test each student? What if there were multiple students? Suggest what facilities you would like to be able to handle this.

Solution: We need to duplicate the nodes for the extra digits. In particular, each extra digit will act just like X_i, Y_1 and Z_1 in how they depend on each other and the previous and the next digits. We need to be able to duplicate the nodes for each problem. But we don't want to duplicate the nodes that don't depend on the actual problem (e.g., the student's knowledge).

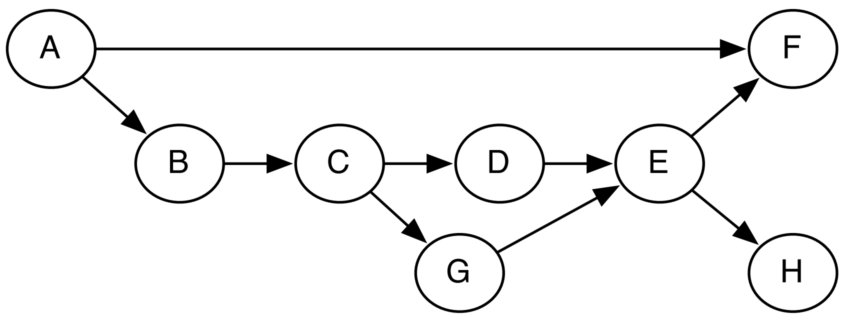
We need a way to duplicate parts of the graph and to connect them to the previous and/or next digits, while preserving the probabilities, and the dependence on other variables.

(d) Suppose we wanted to model teaching the student something. How could we model this? [Hint: think about teaching as a node that specifies what is taught and have separate variables representing before and after teaching. What nodes need to be duplicated?]

Solution: We need to model the students knowledge at different times, and model the effect of teaching actions. We could duplicate the student's knowledge nodes for each time, and they depend in the student's previous knowledge and the teaching activity. We will need to duplicate the Z variables and the carry variables for each time, but not the X and Y variables.

Question 3

Consider the following belief net:



(a) What are all of the factors required to represent this belief network?

Solution: $P(A), P(B|A), P(C|B), P(D|C), P(E|D,G), P(F|E,A), P(G|C), P(H|E)$

(b) Consider the query: $P(G|F = true, H = false)$. What are the factors after the observations have been taked into account?

Solution: $P(A), P(B|A), P(C|B), P(D|C), P(E|D,G), P(f|E,A), P(G|C), P(\neg h|E)$.
Note that $P(f|E,A)$ is a factor on E and A, and not on F.

(c) For that query and for the elimination ordering, B, D, E, A, C give all of factors created in VE. For each factor created, specify which factors were removed, and what variable was summed out. You do not need to consider any numerical values.

Solution:

Var Eliminated	Factors Removed	Factor Created
B	$P(C|B)P(B|A)$	$f_1(A,C)$
D	$P(D|C)P(E|D,G)$	$f_2(C,E,G)$
E	$P(f|E,A)P(\neg h|E))f_2(C,E,G)$	$f_3(A,C,G)$
A	$P(A)f_1(A,C)f_3(A,C,G)$	$f_4(C,G)$
C	$P(G|C)f_4(C,G)$	$f_5(G)$

(d) What is the treewidth of this graph for this elimination ordering? Explain how you know this.

Solution: 3 — this is the size of the largest factor.

(e) Consider using recursive conditioning for the same query, assuming that the variable were split in the order G, C, A, E, D, B. What values are used from the cache in this computation? Do any of the assignments disconnect the graph? If so, which ones?

Solution: From the previous part we see that the factorization is:

$$P(G,f,\neg h) = \sum_C P(G|C) \sum_A P(A) \left(\sum_B P(C|B)P(B|A) \right) \left(\sum_E (f|EA)P(\neg h|E) \sum_D P(D|C)P(E|D,G) \right)$$

The cached values stored after summing out D for one value of A can be used for the other value of A, as the factors do not depend on A. However they will not be used for other values of C, E or G as the values in the factors depend on these variables.

The cached value stored after summing out B can be used for the other value of G, but not for other values of C or A.

After assigning G, C and A, the factors containing $\{P(C = v_1|B), P(B|A = v_2)\}$ are disconnected from the other factors.

Question 4

For each question in this assignment, say how long you spent on it. Was this reasonable? What did you learn?