Assignment 3: Bayesian Inference, Temporal State Estimation and Decision Making under Uncertainty

Alex Smirnov, Scott Reyes

April 11, 2017

Problem 1:

a

The probability that all five of the Boolean variables are simultaneously true is:
$P(A) = 0.2$
$P(B) = 0.5$
$P(C) = 0.8$
$P(D \mid A \wedge B) = 0.1$
$P(E \mid B \wedge C) = 0.3$
$P(A \wedge B) = 0.1$
$P(A \wedge B \wedge C) = 0.08$
$P(A \wedge B \wedge C) \times P(D \mid A \wedge B) = 0.008$
$P(A \wedge B \wedge C) \times P(D \mid A \wedge B) \times P(E \mid B \wedge C) = 0.0024$

b

The probability that all five of the Boolean variables are simultaneously false is:
$P(\neg A) = 0.8$
$P(\neg B) = 0.5$
$P(\neg C) = 0.2$
$P(\neg D \mid \neg A \wedge \neg B) = 0.1$
$P(\neg E \mid \neg B \wedge \neg C) = 0.8$
$P(\neg A \wedge \neg B) = 0.4$
$P(\neg A \wedge \neg B \wedge \neg C) = 0.08$
$P(\neg A \wedge \neg B \wedge \neg C) \times P(\neg D \mid \neg A \wedge \neg B) = 0.008$
$P(\neg A \wedge \neg B \wedge \neg C) \times P(\neg D \mid \neg A \wedge \neg B) \times P(\neg E \mid \neg B \wedge \neg C) = 0.0064$

c

$P(\neg A) = 0.8$
$P(D \wedge B) = 0.7$
$P(D \wedge B \mid \neg A) = 0.6$
$P(\neg A \mid D \wedge B) = \frac{0.8*0.6}{0.7} = 0.686$

Problem 2:

a

Query: $P(Burglary \mid JohnCalls = true, MaryCalls = true)$
Variable Elimination
Query expression:
$P(B \mid j, m) = \alpha f_1(B) * \sum_e f_2(E) * \sum_a f_3(A, B, E) * f_4(A) * f_5 * (A)$
$f_6(B, E) = \sum_a f_3(A, B, E) * f_4(A) * f_5(A)$
$= (f_3(a, B, E) * f_4(a) * f_5(a)) + (f_3(\neg a, B, E) * f_4(\neg a) * f_5(\neg a))$
$P(B \mid j, m) = \alpha f_1(B) * \sum_e f_2(E) * f_6(B, E)$
$f_7(B) = \sum_e f_2(E) * f_6(B, E)$
$= f_2(e) * f_6(B, E) + f_2(\neg e) * f_6(B, \neg e)$
$P(B \mid j, m) = \alpha f_1(B) * f_7(B)$

b

Variable Elimination Algorithm - Arithmetic Operations Performed
Additions: 1
Multiplications: 5
Divisions: 1
Tree Enumeration Algorithm - Operations Performed
Additions: 3
Multiplications: 9
Divisions: 1

c

If a Bayesian network has the form of a chain, the complexity of computing $P(X_1 \mid X_n = true)$ using enumeration is $O(n)$ because every single X would need to be used in the calculation.
Computing the complexity with variable elimination is also $O(n)$ because no variables would be eliminated. Since the Bayesian network is a chain, there would be no eliminated variables and the complexity would be exactly the same as tree enumeration.

Problem 3:

a

Prove:

$$P(X \mid MB(X)) = \alpha P(X \mid U_1, ..., U_m) \prod_{Yi} P(Y_i \mid Z_i, ...)$$

Definition of Bayesian network:

$$(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

via product rule:

$$(x_1, x_2, ..., x_n) = P(x_1 \mid x_{n-1}, ..., x_1) P(x_{n-1}, ..., x_1)$$

Definition of Markov Assumption:

$$P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$$

Using the Markov assumption on the Bayesian network lets all nodes be conditionally independent from the other nodes in the graph given the Markov blanket because X's children conditionally depend on X as well as the child's parents, and x depends on its parents.

b

$$P(Rain \mid Sprinkler = true \wedge WetGrass = true)$$

MCMC would solve this by trying fixing sprinkler and wet grass to true while testing rain by calculating the probability repeatedly randomly changing the non fixed variable values. In the case above, there would be 4 states to take into consideration. Cloudy=T/F, Rain=T/F.

c

Problem 4:

a

Expected net gain from buying C_1 given no test:
$q^+ = 1000$
$q^- = -400$
$P(q^+) = 0.7$
$P(q^-) = 0.3$
$E(X) = (1000 * 0.7) + (-400 * 0.3) = 580$

b

$P(pass(c_1) \mid q^+(c_1)) = 0.8$
$P(pass(c_1) \mid q^-(c_1)) = 0.35$

$P(Pass) = (0.8 * 0.7) + (0.35 * 0.3) = 0.665$
$P(q^+ \mid Pass) = \frac{0.8*0.7}{0.665} = 0.84$
$P(q^- \mid Pass) = \frac{0.35*0.3}{0.665} = 0.16$
$P(q^+ \mid \neg Pass) = 0.16$
$P(q^- \mid \neg Pass) = 0.84$

c

The best decision is to buy the car if it passes the test, and to not buy the car if it fails the test. $U(Pass) = (0.84 * 1000) - 100 = 740$
$U(\neg Pass) = (0.16 * -400) - 100 = -164$

d

The value of optimal information is the quality of the car. Taking C_1 to the mechanic is the better choice because losses are much lower if the car turns out to be bad quality, but the gain if the car is good quality is still very high.

Problem 5 - Programming Component:

a

b

c - Generating Ground Truth Data

d - Filtering and Viterbi Algorithms in Large Maps

e

f

g

h - Computational Approximations