MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.034—Artificial Intelligence—Fall 1996

Solutions **6** Handout **#18**

1. Computing conditional probabilities (5 pts)

```
(define (cp net target given)
  (/ (infer net (append target given))
     (infer net given)))

(cp mcbn1 '((e #t)) '((b #f)))
=> .19692307692307692
(cp mcbn1 '((d #t) (e #t)) '((a #f) (b #f)))
=> .043
(cp mcbn1 '((a #t)) '((d #f) (e #f)))
=> .12336892052194545
```

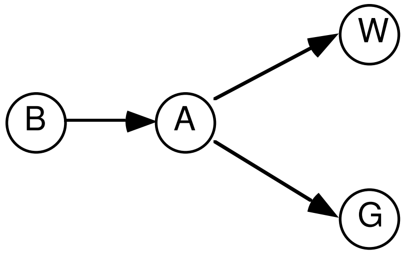
2. Modeling complex probabilistic dependence

Holmes is worried that his house will be burgled. For the time period of interest, there is a 10^{-4} *a priori* chance of this happening, and Holmes has installed a burglar alarm to try to forestall this event. The alarm is 95% reliable in sounding when a burglary happens, but also has a false positive rate of 1%. Holmes' neighbor, Watson, is sure (well, say 90% sure) to call Holmes at his office if the alarm sounds, but he is also a bit of a practical joker, and knowing Holmes' concern, might (30%) call even if the alarm is silent. Holmes' other neighbor, Mrs. Gibbons, is a well-known lush and often befuddled, but Holmes believes that she is four times more likely to call him if there is an alarm than if not.

2.A Building the model (4 pts)

The network to describe Holmes' initial problem formulation involves four binary random variables, and thus four nodes in the Bayes network:

B, whether Holmes' house has in fact been burgled,
A, whether his alarm sounded,
W, whether Watson telephones, and
G, whether Gibbons telephones.



We may reasonably assume that B depends probabilistically only on the intent of some burglar to break in, and we model this only by giving an *a priori* probability of the burglary, 10^{-4}. It also seems reasonable to model the alarm as depending only on whether the burglary actually took place. If it did, the alarm would go off 95% of the time; if it did not, the alarm would still go off 1% of the time. We may also assume that both Watson's and Gibbons' telephoning depends only on the alarm, and not on the burglary itself; i.e., that G and W get their knowledge about the burglary only through whether there is an alarm or not. We also assume that G and W are conditionally independent; e.g., given that the alarm sounds, Gibbons is neither more nor less likely to call based on whether Watson calls or not, and similarly for the case where the alarm does not sound. As the problem set noted, the only use we make of Gibbons' call is to update our belief in whether the alarm sounded, and for this purpose, an almost arbitrary assignment of conditional probabilities for Gibbons is OK so long as the likelihood ratio, P(G|A)/P(G|~A) is 4, as specified.[1]

We can express these in the following tables:

B = #t	*B* = #f
.0001	.9999

A	*W = #t*	*W = #f*
#t	0.90	0.10
#f	0.30	0.70

B	*A = #t*	*A = #f*
#t	0.95	0.05
#f	0.01	0.99

A	*G = #t*	*G = #f*
#t	0.40	0.60
#f	0.10	0.90

```
(define holmes
  (define-bnet 'holmes
    '((b () (0.9999 0.0001))
      (a (b) ((0.99 0.01) (0.05 0.95)))
      (w (a) ((0.70 0.30) (0.10 0.90)))
      (g (a) ((0.90 0.10) (0.60 0.40))))))
```

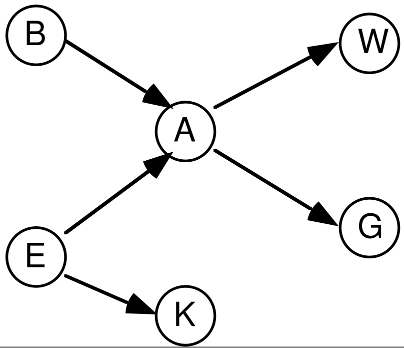
2.B Calculations (2 pts)

```
(cp holmes '((b #t)) '((w #t) (g #t)))
=> 1.0305715216636035e-3
```

Getting the calls from Watson and Gibbons raises Holmes' estimate of the probability of the burglary from 0.0001 to 0.0013, a factor of 13.

[1] If we were trying to answer other questions as well, this arbitrariness would not be appropriate. For example, if I were to ask how likely Gibbons is to call in case the burglary happens, we would need a true estimate of P(G|A). If we wished to use Gibbons' *not calling* to update the probability of the alarm sounding, we would need the likelihood ratio P(~G|A)/P(~G|~A) as well, and this too would require us to make accurate estimates of the conditional probabilities, not just their ratios.

2.C Explaining away (3 pts)



We add nodes to the network above to represent the presence or absence of the earthquake (E) and the radio report (K). In this network, we have a node (the alarm) that now depends on two parent nodes, not just one. In general, we could list all of the conditional probabilities in a table such as the following, which enumerates the probability of alarm under all possible combinations of values of burglary and earthquake:

B	E	A = #t	A = #f
#f	#f	0.01	0.99
#f	#t	0.60	0.40
#t	#f	0.95	0.05
#t	#t	0.98	0.02

If we assume that burglary and earthquake can each independently cause the alarm to sound and that there is no specific synergy or anti-synergy between them, then the noisy-or assumption lets us specify this table in a more compact form. Because we heard in the original story that in the absence of a burglary the alarm still sounds with 1% probability, this number must be the "leak" in the noisy-or model. This implies that the causal probability from burglary to alarm is $P_c(A|B) = 1 - (1-P(A|B))/(1-\text{leak}) = 1 - (0.05/0.99) = 0.949$. (Note that this is very close to the conditional probability, 0.95.) We are told that the causal probability of earthquake causing the alarm is 0.6. Thus, the proper model for the dependence of the alarm on burglary and alarm is:

```
(noisy-or 0.01 0.949 0.6)
```

The overall model that includes the possible effects of the earthquake and newsreports is:

```
(define quake
  (define-bnet
    'quake
    '((b () (0.9999 0.0001))
      (e () (0.9999 0.0001))
      (a (b e) (noisy-or 0.01 0.949 0.6))
      (w (a) ((0.70 0.30) (0.10 0.90)))
      (g (a) ((0.90 0.10) (0.60 0.40)))
      (k (e) ((1.0 0.0) (0.05 0.95))))))
```

and the following conditional probabilities show the likelihood of a burglary under various conditions. The first closely matches the result in the model above because earthquakes are generally unlikely. The second shows that knowing the quake actually did not occur makes little difference, for the same reason. The third gives the required answer: knowing (from the radio) that the quake occured, the probability of burglary declines by an order of magnitude, because it can explain the alarm without needing to hypothesize the burglary.

```
: (cp quake '((b #t)) '((w #t) (g #t)))
1.0294845035286652e-3
: (cp quake '((b #t)) '((w #t) (g #t) (e #f)))
1.0300868877251574e-3
: (cp quake '((b #t)) '((w #t) (g #t) (k #t)))
1.5407125972589493e-4
```

2.D Multiple connections (3 pts)

We model Holmes' daughter's calling by a node D, which depends on A and E. We asked you to make up some "reasonable" model for this. Here is one interpretation:

A	E	D = #t	D = #f
#f	#f	0.01	0.99
#f	#t	0.00	1.00
#t	#f	0.98	0.02
#t	#t	0.01	0.99

This leads to an overall network:

```
(define daughter
  (define-bnet
    'daughter
    '((b () (0.9999 0.0001))
      (e () (0.9999 0.0001))
      (a (b e) (noisy-or 0.01 0.949 0.6))
      (w (a) ((0.70 0.30) (0.10 0.90)))
      (g (a) ((0.90 0.10) (0.60 0.40)))
      (k (e) ((1.0 0.0) (0.05 0.95)))
      (d (a e) (((0.99 0.01) (1.00 0.00))
                ((0.02 0.98) (0.99 0.01)))))))
```

and the requested probability, that there is a burglary given no call from the daughter, is 0.000007:

```
: (cp daughter '((b #t)) '((d #f)))
7.046184549388095e-6
```

Note that the daughter's negative testimony is quite telling. Even if Watson and Gibbons' testimonies are positive, the posterior probability of the burglary is still lower than its a priori value.

```
: (cp daughter '((b #t)) '((d #f) (w #t) (g #t)))
2.8380531740652885e-5
```

3. Probabilistic diagnosis

3.A Defining a bipartite graph Bayesian problem (3 pts)

```
(define dom3b
  (define-bnet 'dom3b
    '((a () (.9 .1))
      (b () (.98 .02))
      (c () (.95 .05))
      (d () (.85 .15))
      (e () (.9 .1))
      (f () (.8 .2))
      (g () (.9 .1))
      (h () (.99 .01))
      (s1 (a b c d) (noisy-or .05 .8 .8 .7 .9))
      (s2 (a b e) (noisy-or .05 .7 .8 .9))
      (s3 (c e g h) (noisy-or .05 .85 .8 .7 .7))
```

```
      (s4 (b c e g) (noisy-or .05 .9 .7 .7 .7))
      (s5 (a c d f h) (noisy-or .05 .8 .8 .8 .8 .8)))))
```

Unfortunately, the graph showing these causal probabilities in the assignment was unclear. The above numbers were the intended values, but consistent solutions with other interpretations will also be accepted.

3.B Exploring absent symptoms (2 pts)

```
: (cp dom3b '((a #t)) '((s1 #t) (s2 #t) (s3 #t)))
.45140550971765836

: (cp dom3b '((a #t)) '((s1 #t) (s2 #t) (s3 #t) (s4 #f) (s5 #f)))
.36589535918010524
```

Unlike in the diagnostic systems based on set cover, the *absence* of symptoms matters in probabilistic systems. Learning that s4 and s5 are both false decreases the likelihood of A to about 9% less than if we simply knew nothing about these variables. This seems much more reasonable than the behavior of the set cover algorithm.

3.C Joint probabilities of multiple diseases (2 pts)

```
: (cp dom3b '((e #t)) '((s1 #t) (s2 #t) (s3 #t)))
.6309444233918594

: (cp dom3b '((h #t)) '((s1 #t) (s2 #t) (s3 #t)))
.0229411100551547

: (* (cp dom3b '((a #t)) '((s1 #t) (s2 #t) (s3 #t)))
     (cp dom3b '((e #t)) '((s1 #t) (s2 #t) (s3 #t)))
     (cp dom3b '((h #t)) '((s1 #t) (s2 #t) (s3 #t))))
6.533898597480341e-3

: (cp dom3b '((a #t) (e #t) (h #t)) '((s1 #t) (s2 #t) (s3 #t)))
2.2194334431509927e-3
```

Ben is, as usual, wrong. One way to understand why the joint probability is lower than the product of the individual ones is the following: the individual probabilities measure the total likelihood that that disease is present under all other possible sets of circumstances. When we seek the joint probability, we accept only cases where the disease is present in the presence of the others.

3.D Use for diagnosis (3 pts)

Just from examining the bipartite graph, diseases A and E appear attractive because each predicts two of the three observed symptoms. B suffers from a lower *a priori* probability. Alas, each also predicts one symptom that is known to be absent and each fails to predict one that is present. We can compute the probability using our method for computing conditional probabilities. For example, this is the probability that only the disease A is present given that s1, s2 and s3 are present and s4 and s5 are absent:

```
: (cp dom3b '((e #f) (d #f) (c #f) (b #f) (a #t))
      '((s1 #t) (s2 #t) (s3 #t) (s4 #f) (s5 #f)))
.14243996471290132
```

To reduce typing, we define a utility that generates the arguments to cp from the list of diseases hypothesized to be present. Note that the absence of a disease from the argument to try is taken to mean that it is *absent*, not *don't care*.

```
(define (try l)
  (cp dom3b
      (map (lambda (dis)
             (list dis (if (memq dis l) #t #f)))
           '(a b c d e f g h))
      '((s1 #t) (s2 #t) (s3 #t) (s4 #f) (s5 #f))))

: (try '(a))
.14243996471290132

: (try '(e))
.27043671622064847
```

Adding additional diseases to either of the explanations A or E may be worth exploring. We can get additional coverage of observed symptoms, but we also pay the "penalty" of predicting more unobserved symptoms. Again, by inspection of the graph, we can try {A E}, {D E}:

```
: (try '(a e))
.10451109075249669

: (try '(d e))
.17276133753860226
```

In this particular situation, the multi-disease hypotheses turn out to be less likely than single-disease ones. This is because their *a priori* probabilities are low and because each additional disease also predicts other, unobserved, symptoms. If we had a different set of symptoms, the results would be otherwise. For example, if we had observed all symptoms but `s1`, the combination of diseases {E F} would look more attractive than just E by itself.

```
: (cp dom3b '((h #f) (g #f) (f #f) (e #t) (d #f) (c #f) (b #f) (a #f))
      '((s1 #f) (s2 #t) (s3 #t) (s4 #t) (s5 #t)))
.11278717046417333

: (cp dom3b '((h #f) (g #f) (f #t) (e #t) (d #f) (c #f) (b #f) (a #f))
      '((s1 #f) (s2 #t) (s3 #t) (s4 #t) (s5 #t)))
.4567880403799016
```

4. Grand Synthesis

4.A Design (3 pts)

The trick is to find an appropriate subset of the set of all possible hypotheses for which it is worthwhile to evaluate the joint probability that just those diseases are present given the set of observed and unobserved symptoms. The most straightforward idea is to use Reggia's set cover algorithm to generate the minimal cardinality diagnoses for the observed symptoms, then use the Bayes network to evaluate the likelihood of each of those diagnoses. This is of course not guaranteed to find the best (maximum likelihood) solution because there is no guarantee that the most likely solution is in fact among the smallest covers, and because Reggia's algorithm is itself incomplete (it may miss the best cover).

As an elaboration, we could use the hypotheses suggested by the set cover algorithm as starting points for a limited search in hypothesis space. We might consider, for example, hypotheses obtained from the starting ones by adding one more disease to each or by deleting one from multi-disease hypotheses. This approach would take extra computation but would be likely to find higher-probability solutions that are "close" to those suggested by set cover.

4.B Implementation (Extra Credit of 5 pts)

The official solutions do not include an implementation.