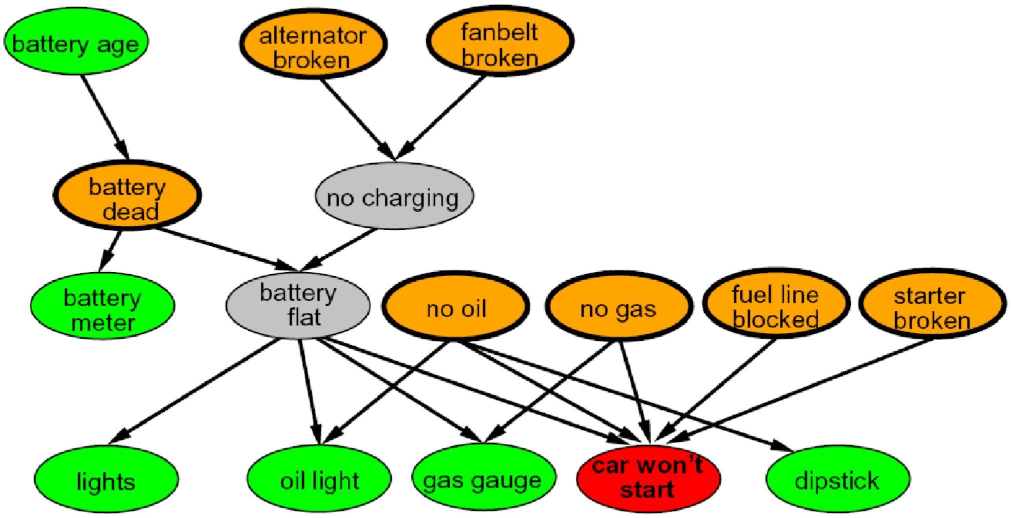
This document will cover two aspects:

1. The values to be attached to the probability tables, and
2. The formulae to be used to compute the probabilities that are required.

Probability table specification

In total there are 16 nodes for the Car diagnosis Bayesian network. This means that 16 probability tables are required. These tables must be assigned values according to the specifications below.



1. Battery age table

ba _y	0.2
ba_n	0.8

2. Alternator broken table

ab_y	0.1
ab_n	0.9

3. Fanbelt broken table

fb_y	0.3
fb_n	0.7

4. Battery dead table

ba_y_bd_y	0.7
ba_y_bd_n	0.3
ba_n_bd_y	0.3
ba_n_bd_n	0.7

Note that this table represents conditional probabilities. Thus for example, the row ba_y_bd_y = 0.2 should be interpreted as the Pr(battery being dead|battery is aged) is 0.2. All other tables below which have more than two rows also happen to represent conditional probabilities.

5. No charging table

ab_y_fb_y_nc_y	0.75
ab_y_fb_n_nc_y	0.4
ab_n_fb_y_nc_y	0.6
ab_n_fb_n_nc_y	0.1
ab_y_fb_y_nc=n	0.25
ab_y_fb_n_nc=n	0.6
ab_n_fb_y_nc=n	0.4
ab_n_fb_n_nc=n	0.9

6. Battery meter table

bd_y_bm_y	0.9
bd_y_bm_n	0.1
bd_n_bm_y	0.1
bd_n_bm_n	0.9

7. Battery flat table

bd_y_nc_y_bf_y	0.95
bd_y_nc_n_bf_y	0.85
bd_n_nc_y_bf_y	0.8
bd_n_nc_n_bf_y	0.1
bd_y_nc_y_bf_n	0.05
bd_y_nc_n_bf_n	0.15
bd_n_nc_y_bf_n	0.2
bd_n_nc_n_bf_n	0.9

8. No oil table

no _y	0.05
no_n	0.95

9. No gas table

ng _y	0.05
ng_n	0.95

10. Fuel line blocked table

flb _y	0.1
flb_n	0.9

11. Starter broken table

sb _y	0.1
sb_n	0.9

12. Lights table

l_y_bf_y	0.9
l_n_bf_y	0.1
l_y_bf_n	0.3
l_n_bf_n	0.7

13. Oil lights table

no_y_ol_y_bf_y	0.8
no_n_ol_y_bf_y	0.05
no_y_ol_n_bf_y	0.05
no_n_ol_n_bf_y	0.1
no_y_ol_y_bf_n	0.8
no_y_ol_n_bf_n	0.05
no_n_ol_y_bf_n	0.05
no_n_ol_n_bf_n	0.1

14. Gas gauge table

ng_y_gg_y_bf_y	0.90
ng_y_gg_n_bf_y	0.05
ng_n_gg_y_bf_y	0.1
ng_n_gg_n_bf_y	0.05
ng_y_gg_y_bf_n	0.8
ng_y_gg_n_bf_n	0.05
ng_n_gg_y_bf_n	0.05
ng_n_gg_n_bf_n	0.1

15. Car won't start table

This table has 64 rows. There are 3 cases to consider:

1. For every combination of bf, no, ng, fb, sb with at least one of these variables taking the value y, the probability is 0.9 for the cs_n outcome.
2. For the case when all 5 variables bf, no, ng, fb, sb take the value n with cs_n. In this case the probability is 0.1

3. The remaining case is when cs_y. The probabilities are now defined as the complement of the probabilities of the first 32 rows. That is if the probability is p for the first row then it is (1-p) for the 33^{rd} row, if it is q for row 2 then it is (1-q) for row 34 and so on.

Note this table must be encoded in the graph with 64 rows and each row should have a probability as specified above.

16. Dipstick low table

no_y_dl_y	0.95
no_n_dl_y	0.3
no_y_dl_n	0.05
no_n_d_n	0.7

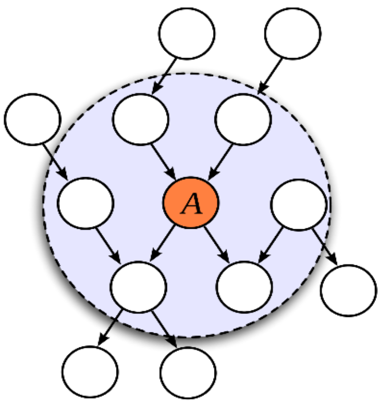
Formulae for computation of probabilities

In addition to the discussion below you are strongly advised to refer to the class notes on Bayesian learning.

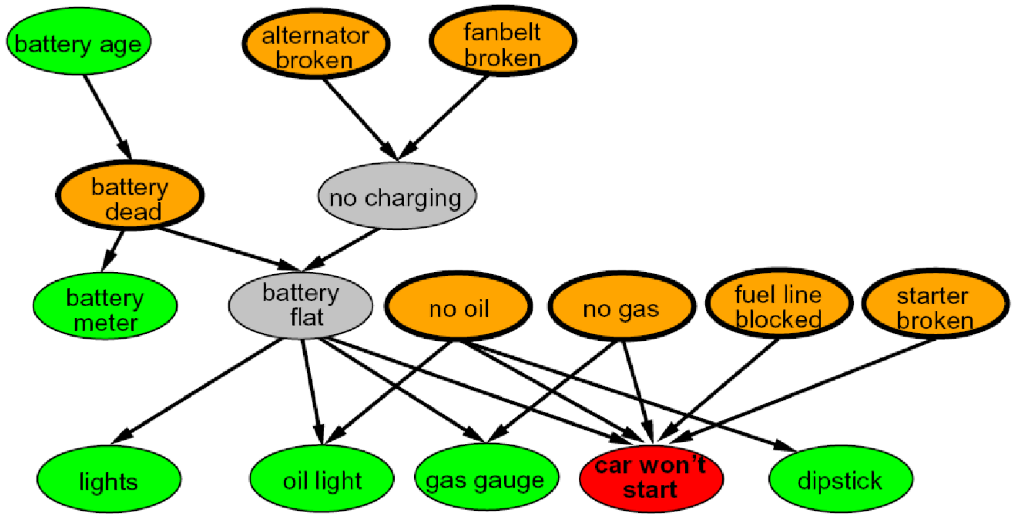
Let us illustrate the computation of the probabilities by taking R2 as an example.

For R2 you are asked to compute P(-cs, +ab, +fb) - this is the joint probability that the car does not start whenever both the alternator and fan belt are not functioning at the same time. This probability involves using the bf variable. We need to first understand the "span" or "scope" of this variable. The span of a variable is the set of variables that it influences together with the variables that influence it.

The span of a variable is given by its Markov blanket.



In the figure above the Markov blanket of variable A is the set of variables in the circle surrounding variable A. These variables are 1) the parents of A, 2) the children of A, and 3) the other parents (i.e., the ones not included in step 1) of children of A.



Applying the Markov blanket principle to the bf variable, we get its Markov blanket (M) as: M= {bd, nc, bf, l, ol, gg, cs, dl, no, ng, flb, sb}.

We see that M contains 12 elements and only 1 of them (cs) is directly relevant and another (nc) is indirectly relevant to the query P(-cs, +fb, +ab). So, what do we do about the other 10 variables? The answer is that we account for the effect of these variables by a process of *enumeration.* Enumeration simply means that we take arithmetic sums of probability over all possible values of the variables that are not relevant to the query.

We also note that P(-cs|+ab, +fb) = P(-cs|+nc) as only the immediate parent of bf (which is nc) is sufficient to evaluate probabilities of its grandparents ab and fb.

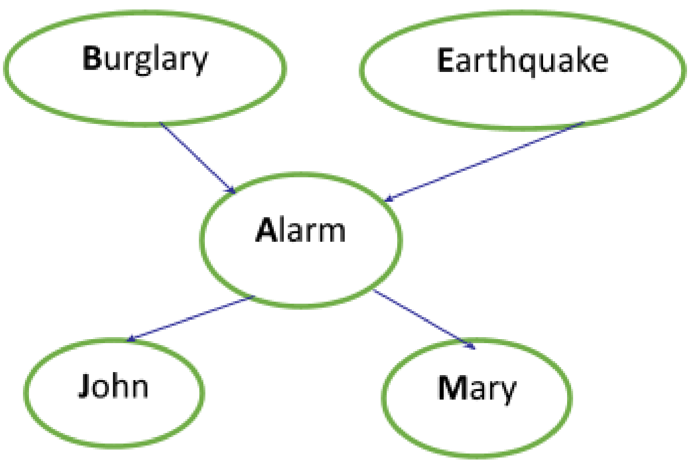
We now have P(-cs|+ab, +fb) = $\sum_{l,ol,gg,no,ng,flb,sb} P(-cs,\ +nc,\ +bf,\ l,\ ol,\ gg,\ no,\ ng,\ flb,\ sb,\ +bd)$

$$= \sum_{l,ol,gg,no,ng,flb,sb} P(+bf|(+nc, + bd) \times P(l| + bf) \times P(ol| + bf,\ no) \times P(gg| + bf,\ ng) \times P(-cs| + bf,\ no,\ n$$

-----(1)

Since there are 7 free variables (i.e., variables without a + or - sign in front of them) there are a total of 2^7=128 possibilities and hence there will be 128 sums in equation 1.

Here is a simpler example to understand the use of summation over free variables. Suppose that we are interested in computing P(B|+j,+m) for the example below



$$P(B| +j, +m) = \sum_{e,a} P(B,\ e,\ a,\ \ +j,\ \ +m)$$

$$= \sum_{e,\,a} P(B)\times P(e)\times P(B,\ e)\times P(a)\times P(a)$$

$$= P(B)\times P(+\ e)\times P(B,\ \ +e)\times P(+\ a)\times P(+\ a) + P(B)\times P(+\ e)\times P(B,\ \ +e)\times P(-\ a)\times P(-\ a) + P(B)\times P(-\ e)\times$$

as we have 4 combinations of e and a, which are (1) +e, +a (2) +e, -a (3) -e, +a (4) -e, -a.

For programming this expression, we need two for loops, the outer one runs for variable *e* and the inner loop that runs for variable *a*. Each for loop needs to take two values, one value for the (+) case and one value for the (–) case.

For answering R2, we have 7 free variables that can be programmed by having 7 nested for loops, one loop for each of the variables.