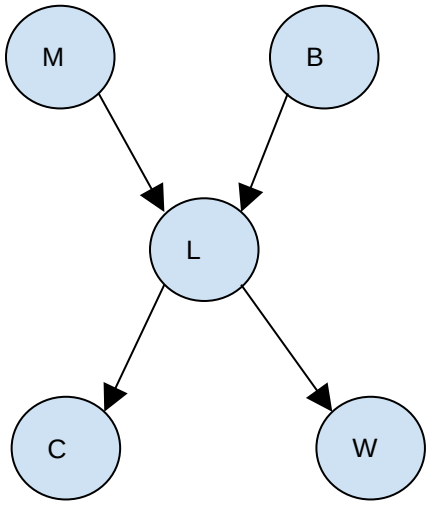
The chance that I arrive late to work is dependent on two factors: my morning routine being miserable, and whether the bus is delayed to my stop. The time that I arrive at work influences whether I get a coffee; it also influences whether I decide to stay late at work in the evening. (Inference question: suppose it's past 6pm, and you wanted to ask me a question about an assignment; you saw me getting a coffee in the morning. what is the chance that I am still in my office?)

Step 1: Create random variables, e.g., Miz Morning (M), Bus Delay (B), I'm Late (L), Coffee (C), and Work Late (W).

Step 2: Connect variables that directly impact each other, then specify conditional probability tables (CPTs) for each node given the topology, e.g.:



MizMorning:M	**P**(M)
true	.9
false	.1

BusDelay:B	**P**(B)
true	.4
false	.6

I'mLate:L	**P**(L|M=true,B=true)	**P**(L|M=true,B=false)	**P**(L|M=false,B=true)	**P**(L|M=false, B=false)
true	.8	.7	.6	.1
false	.2	.3	.4	.9

Coffee:C	**P**(C | L=true)	**P**(C | L=false)
true	.3	.6
false	.7	.4

WorkLate:W	**P**(W | L=true)	**P**(W | L=false)
true	.4	.2
false	.6	.8

We can build the full joint probability distribution for **P**(M,B,L,C,W), which will allow us to make quantitative inferences. We do this by multiplying the appropriate conditional probabilities by visiting each node; we'll traverse the structure starting at the roots, going from parent nodes to child nodes.

P(M,B,L,C,W) = **P**(M) **P**(B) **P**(L|M, B) **P**(C | L) **P**(W|L)

Then we can reason, e.g.

Pr(M=true,B=true,L=true,C=false,W=true)
= Pr(M=true)Pr(B=true)Pr(L=true|M=true, B=true)Pr(C=false | L=true)Pr(W=true|L=true)
= 0.9*0.4*0.8*0.7*0.4

Another example: If we know there is a bus delay, how often do I stay late vs not?

P(W | B=true) = **P**(W, B=true) / Pr(B=true). Consider the numerator:
P(W, B=t)
= $\text{sum}_{l=\{t/f\}}$ $\text{sum}_{c=\{t/f\}}$ $\text{sum}_{m=\{t/f\}}$ **P**(W, B=t, L=l, C=c, M=m)
= $\text{sum}_{l=\{t/f\}}$ $\text{sum}_{c=\{t/f\}}$ $\text{sum}_{m=\{t/f\}}$ Pr(M=m)Pr(B=t)Pr(L=l|M=m,B=t)Pr(C=c|L=l)**P**(W|L=l)

Below, we enumerate each term of the summation:

l	c	m	**P**(W, B=t, L=l, C=c, M=m)
t	t	t	Pr(B=t)Pr(M=t)Pr(L=t|M=t,B=t)Pr(C=t|L=t)**P**(W|L=t) [0.9*0.4*0.8*0.3*0.4, 0.9*0.4*0.8*0.3*0.6]
t	t	f	Pr(B=t)Pr(M=f)Pr(L=t|M=f,B=t)Pr(C=t|L=t)**P**(W|L=t) [0.1*0.4*0.6*0.3*0.4, 0.1*0.4*0.6*0.3*0.6]
t	f	t	Pr(B=t)Pr(M=t)Pr(L=t|M=t,B=t)Pr(C=f|L=t)**P**(W|L=t) [0.9*0.4*0.8*0.7*0.4, 0.9*0.4*0.8*0.7*0.6]
t	f	f	Pr(B=t)Pr(M=f)Pr(L=t|M=t,B=t)Pr(C=f|L=t)**P**(W|L=t) [0.1*0.4*0.6*0.7*0.4, 0.1*0.4*0.6*0.7*0.6]
f	t	t	Pr(B=t)Pr(M=t)Pr(L=f|M=t,B=t)Pr(C=t|L=f)**P**(W|L=f) [0.9*0.4*0.2*0.6*0.2, 0.9*0.4*0.8*0.6*0.8]
f	t	f	Pr(B=t)Pr(M=f)Pr(L=f|M=f,B=t)Pr(C=t|L=f)**P**(W|L=f) [0.1*0.4*0.4*0.6*0.2, 0.1*0.4*0.4*0.6*0.8]
f	f	t	Pr(B=t)Pr(M=t)Pr(L=f|M=t,B=t)Pr(C=f|L=f)**P**(W|L=f) [0.9*0.4*0.2*0.4*0.2, 0.9*0.4*0.8*0.4*0.8]
f	f	f	Pr(B=t)Pr(M=f)Pr(L=f|M=f,B=t)Pr(C=f|L=f)**P**(W|L=f) [0.1*0.4*0.4*0.4*0.2, 0.1*0.4*0.4*0.4*0.8]

Doing this naively, we'd have to perform 32 multiplications and 7 adds (4 multiplications per product term; we have 8 product terms to add) for each possible outcomes of W (i.e., we have to do this for both W=true and W=false).

In listing out all the product terms, we see that we do a lot of repeated work. For example, in four out of the eight terms, we always multiply Pr(M=true)*Pr(B=true). In fact, since we are given that B=true in the problem, we know that all 8 terms will contain Pr(B=true). We can do less computation if we factor out common terms.

$$\mathbf{P}(W,B=t)=\text{sum}_{l=\{t/f\}}\text{sum}_{c=\{t/f\}}\text{sum}_{m=\{t/f\}}Pr(M=m)Pr(B=t)Pr(L=l|M=m,B=t)Pr(C=c|L=l)\mathbf{P}(W|L=l)$$

$$= Pr(B=t)*[\text{sum}_{m=\{t/f\}}Pr(M=m)*[\text{sum}_{l=\{t/f\}}Pr(L=l|M=m,B=t)*\mathbf{P}(W|L=l)*[\text{sum}_{c=\{t/f\}}Pr(C=c|L=l)]]]$$

Now for the denominator. Note that we can calculate Pr(B=true) easily once we have **P**(W, B=true) because we just have to sum out W: Pr(B=true) = Pr(W=true, B=true) + Pr(W=false, B=true). Although we can work directly from the full joint (summing out W, L, C, M) that would require a lot more work.

Naïve Bayes: Special Case of Bayesian Network

Suppose we want a system to scan our incoming emails and then decide to bin each into one of three categories: work, school, social. Suppose we want to do this with a Bayesian Network. Assume that you have a pretty sizable collection of old emails that have been correctly categorized.

For this example, let's simplify it to a problem of modeling the joint relationship between the category and a small set of N keywords that we think will help us distinguish between the three categories. So we want a network with N+1 nodes/variables. Whatever network we design, it will help us to compute the full joint distribution

P(Category, W_1, … W_N)

from which we will be able to make inferences when given a new email.

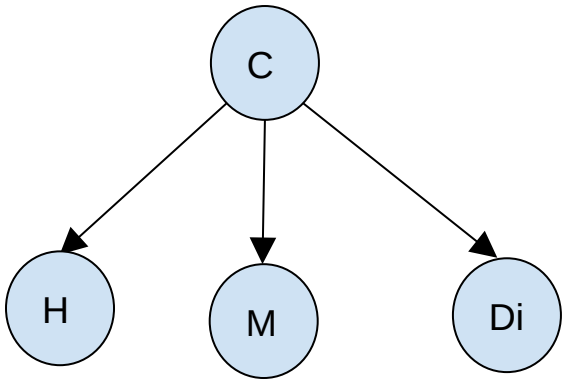
For example, suppose that "homework" and "dinner" are two of the keywords, and suppose both these words appeared in the email we want to categorize, but none of the other N-2 keywords appeared. What we want to infer from the above full joint is:

max c={sch, work, soc} Pr(Category=c | Homework=t, …, Dinner=t)

This can be computed as:

max c={sch, work, soc} Pr(Category=c , Homework=t, …, Dinner=t)

since all three choices of c share the same denominator Pr(Homework=t, … Dinner=t).



Naïve Bayes Model:

The root node is a random variable over the category (work, school, social). It has many children nodes. Each child node is a Boolean random variable, specifying whether the email contains some particular word or not (e.g., in the figure to the left, the first child node is for whether the word "homework" appeared in the email or not, given the category.). All children nodes are conditionally independent of each other. The full joint is computed as:

P(Category, W_1, … W_N) = **P**(Category)*$\mathbf{P}(W_1|\text{Category})$*...*$\mathbf{P}(W_N|\text{Category})$

To use it to answer the specific instance above ("homework" and "dinner" were the only two keywords that appeared in the email), we'd compute:

Pr(Category=sch)*Pr(Homework=t | Category=sch) * … * Pr(Dinner=t | Category=sch)

Pr(Category=soc)*Pr(Homework=t | Category=soc) * … * Pr(Dinner=t | Category=soc)

Pr(Category=work)*Pr(Homework=t | Category=work) *…* Pr(Dinner=t|Category=work)

And we'd categorize the email as whichever one that has the largest probability