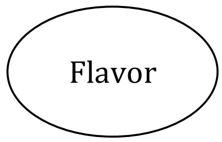
CS B551: Elements of Artificial Intelligence

Enrique Areyan

Homework 4

Answers to Written Questions:

1)

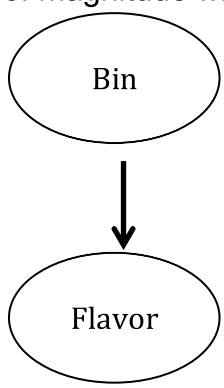


---	P(Flavor)
Lime	0.3
Lemon	0.3
Cherry	0.4

a) Cherry.
b) The fraction of each flavor approximates the true proportion as N increases.
c) The error decreases by several orders of magnitude with an increase in N.

2)

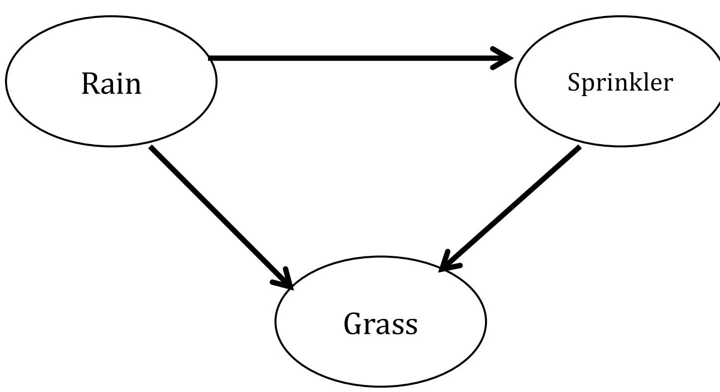
---	L	Li	C
1	0.0	0.0	1.0
2	0.0	1.0	0.0
3	1.0	0.0	0.0
4	0.0	0.33	0.67
5	0.0	0.67	0.33
6	0.33	0.0	0.67
7	0.67	0.0	0.33
8	0.33	0.67	0.0
9	0.67	0.33	0.0
10	0.33	0.33	0.33



---	P(Bin)
1	0.1
2	0.1
…	…
10	0.1

P(Bin | Lime) = {'b10': 0.09909909909909909, 'b4': 0.09909909909909909, 'b5': 0.20120120120120116, 'b6': 0.0, 'b7': 0.0, 'b1': 0.0, 'b2': 0.30030030030030025, 'b3': 0.0, 'b8': 0.20120120120120116, 'b9': 0.09909909909909909}

3) a)



Rain	P(Spri)
T	0.1
F	0.95

---	P(Rain)
T	0.3

Rain	Spri	P(Grass)
T	T	0.95
T	F	0.9
F	T	0.9
F	F	0.0

b) 'Grass',{} = {'nowet': 0.13000000000000003, 'wet': 0.87}

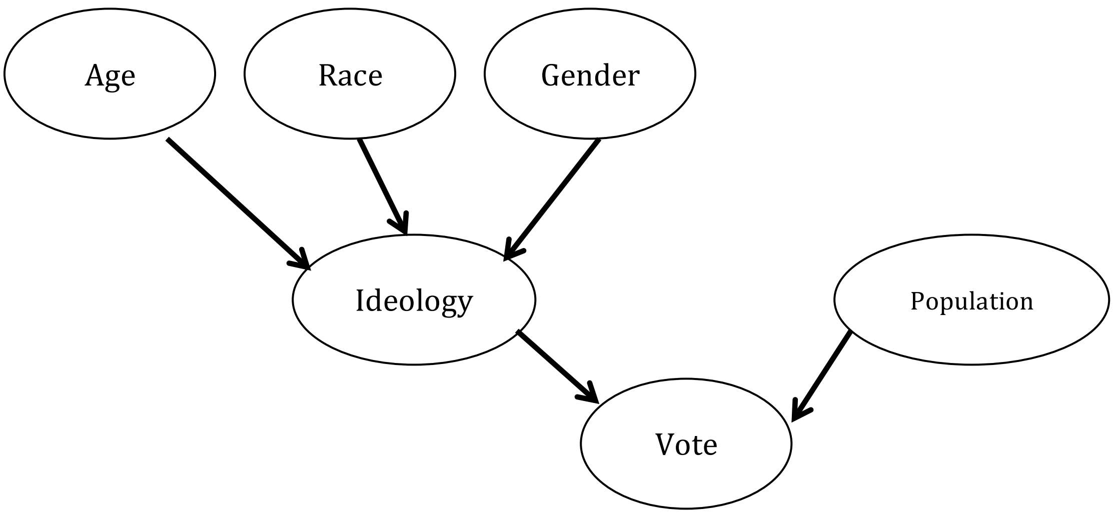
c) 'Rain',{'Grass':'wet'} = {'nr': 0.6879310344827586, 'r': 0.31206896551724145}

d) Two random variables A and B are conditionally independent given C, if:

P(A,B|C) = P(A|C)P(B|C), hence P(A|B,C) = P(A|C). To know whether Rain and Sprinkler are independent of Grass, we would test P(G|R,S) to be equal to P(G|S). If this is true, then there are independent.

We know from the CPT at the network that P(G|R,S) = 0.95. However, P(G|S) = Σ_rP(G,R|S) = Σ_rP(G|R,S)P(R|S) = P(G|R=T,S)P(R=T|S)+P(G|R=F,S)P(R=F|S) = 0.95 x 0.3 + 0.9 x 0.7 = 0.285 + 0.63 = 0.915. Therefore, they are not independ.

4) a)



b) Age ⊥ Race, Age ⊥ Gender, Race ⊥ Gender, Age ⊥ Population, Race ⊥ Population, Gender ⊥ Population, Population ⊥ Ideology.

c) It turns out that by setting approximately uniform, equal probabilities to the CPT on node Ideology and Vote, I obtain an accuracy of 111/10000 = 0.01 or roughly 1%. Probabilities on parent nodes remain fixed as can be seen on my code. Now, when I hand-tune probabilities only on the Vote node, I get an accuracy of 71,27%. When I add hand-tuned probabilities to the Ideology node, I get an accuracy of 69,67%. The greatest difficulties was to find probabilities on combination of data for which I have no idea, e.g., how will the distribution over ideology be on a male, of “other” race, on age scale 1? I have not a clue and must rely only on common stereotypes and my view of the world. Moreover, even common stereotypes do not fill out the whole spectrum, at which point I would just guess some numbers.

d) *ml_result* gives the maximum likelihood estimate which is basically the node with the highest probability among the nodes returned by *enumerate_ask*. In practice, to use the results of *enumerate_ask* one would have to decide among a set of options with different probabilities. To choose the node with the highest probability is the same as using *ml_result*, so in this sense the two are related. Ml_result can be viewed as a strategy to pick one node from the set of possible nodes returned by *enumerate_ask*. However, there can be other strategies, e.g., a roulette-wheel selection where each node will be assigned a portion of the roulette according to each probability.

monte_carlo_estimate relies on multiple, random samples of the CPT and as such approximate *enumerate_ask* in the limit. However, the quality of the approximations depend on the number of samples, so in practice it may be take more calculations to get a set of useful values. Still, one we would have to decide which value to pick out of the set obtained by the estimate.

e) A trained net achieves an accuracy of 72,85%, a slighter better result than my first attempt at hand-tuned net (71,27%) and a more accurate result than my second attempt (69,67%).

Training a net is better than hand-tune it. If one has a sufficient large amount of data available to train a network, one would not have to guess probabilities of potentially enormous tables, a task to tedious and error prone for humans.