Timothy Chisholm

Homework 3

CSCI-630
March 20, 2019

Question 1

RN 6.3

a A greedy best-first search will work well here with the assumption that words of improper length are ignored and with the heuristic being the sum of the frequency of unique letters F minus the frequency of the uniquely constrained letters C. The higher the frequency of the letter means it's more frequently used in the English language, that means it will be easier to find potential cross-word matches for the puzzle. The optimal word will be one that has the highest unique frequency given the size of the word. [1]

b For a CSP the constraint should be on the words because it's not an issue of finding letters that will be the issue, it's that the search will need to look for words with specific letters in specific indices. As the problem goes on the constraint list will grow.

c I believe that a CSP will be better because as the list of constraints grows the number of potential matched words will start to shrink which will make finding one of the right size very fast in the later stages. The search will work well for a while, but toward the end of the problem the search will have to compute much larger trees with more variables.

References

[1] http://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html

Question 2

RN 13.3

a $P(a|b,c) = P(b|a,c) \implies \frac{P(a,b,c)}{P(b,c)} = \frac{P(a,b,c)}{P(a,c)} \implies a = b$
if $a = b$ then $P(a|c) = P(b|c)$
QED

b $P(a|b,c) = P(a) \implies P(b|c) = P(b)$
$\frac{P(a,b,c)}{P(b,c)} = P(a)$
$\frac{P(b,c)}{P(c)} = P(b) \implies P(b,c) = P(b)P(c)$
$\frac{P(a,b,c)}{P(b)P(c)} = P(a) \implies P(a,b,c) = P(a)P(b)P(c)$
QED

c $P(a|b) = P(a) then P(a|b,c) = P(a,c)$
$P(a|b) = \frac{P(a,b)}{P(b} = P(a) \implies P(a,b) = P(a)P(b) \implies a \perp b$
$P(a|b,c) = P(a,c) \implies \frac{P(a,b,c)}{P(b,c)} = P(a,c) \implies P(a,b,c) = P(a,c)P(b,c) \implies$ $(a,c) \perp (b,c) \implies a \perp b$ which is consistent with findings.
QED.

Question 3

RN 13.10

a Expected return is 0.39 : 1

b Probability of a winning spin is 0.375

c Mean : 37
Median: 10

Question 4

RN 18.6 GAIN() Calculations...

A_1 $0.97 - [\frac{4}{5}B(\frac{2}{4}) + \frac{1}{5}B(\frac{0}{1})] =$
$B(\frac{2}{4}) = B(0.5) = 1$
$B(\frac{0}{1}) = B(0) = 0$
$0.97 - [\frac{4}{5} * 1] = 0.97 - 0.80 = 0.17$

A_2 $0.97 - [\frac{3}{5}B(\frac{2}{3}) + \frac{2}{5}B(\frac{0}{2}) =$
$B(\frac{2}{3}) = B(0.667) = 0.918$
$B(\frac{0}{2}) = B(0) = 0$
$0.97 - [\frac{3}{5} * 0.917] = 0.97 - 0.55 = 0.42$

A_3 $0.97 - [\frac{2}{5}B(\frac{1}{2}) + \frac{3}{5}B(\frac{1}{3}) = 0$ $B(\frac{1}{2}) = B(0.5) = 1$
$B(\frac{1}{3}) = B(0.333) = 0.918$
$0.97 - [\frac{2}{5} * 1 + \frac{3}{5} * 0.918] = 0.97 - [0.4 + 0.55] = 0.02$

Choice
1 A_2 with $GAIN(A_2) = 0.42$
2 A_1 with $GAIN(A_1) = 0.17$
3 A_3 with $GAIN(A_3) = 0.02$

Question 5

R&N 18.9

a
```
function DECIDE-R(decision_subtree, example, examples, _weight
    if decision_subtree.children = ∅
        return decision_subtree.root, _weight
    else
        test = decision_tree.root
        options = decision_tree.children
        test_value = example[test]
        if test_value = ∅
            weights, values = [sum(e ∈ examples and e.A = tes
            for v_k in values
                weight = weights[k] * _weight
                DECIDE-R(decision_tree.children[v_k], example,
        else
            DECIDE-R(decision_tree.children[test_value], examp
```

b
```
function W-INFORMATION-GAIN(a, examples) returns gain
    N = |examples|
    T = {e ∈ examplesande.A ≠ ∅}
    gain
    for each value v_k of T:
        weight = (T_k/N)
        gain += INFORMATION-GAIN(a, examples) * weight
    return gain
```