CS 4700: Foundations of Artificial Intelligence

Review questions for Quiz 2

For each of the following problems assume g(N)=$\sqrt{\log_{10} N}$.

1. Consider a multi-armed bandit problem with four arms, 1, 2, 3, and 4, each of which return a positive-valued reward. Imagine there have been 7 prior arm pulls – 2 pulls for each of arms 1, 2, and 3, and 1 pull for arm 4. Further, given the values of the rewards received up through that point, the UCB heuristic says to pull arm 4 as the 8th arm pull. After the N=8 arm pulls the relevant statistics are:
   - Sum_1=1 , N_1=2
   - Sum_2=2, N_2=2
   - Sum_3=3, N_3=2
   - Sum_4=4, N_4=2

   What is the smallest and largest values of the reward that arm 4 could ever have returned for its first pull?

   Since the 8th pull involved arm 4, the statistics for arms 1-3 after 7 pulls would be the same after 8 pulls – none of the three got pulled this 8th time:
   - Sum_1=1 , N_1=2
   - Sum_2=2, N_2=2
   - Sum_3=3, N_3=2

   Furthermore, let's use a variable, V, to refer to the value that arm 4 had at that point:
   - Sum_4=V, N_4=1

   Since N_4=1 that means we're saying that the value after its first pull was V – that's the quantity we're interested in.

   Flashing ahead to just before the 8th arm pull, let's see what each arm's UCB value is:
   - Arm 1: $\frac{1}{2} + \frac{\sqrt{\log_{10} 7}}{\sqrt{2}} = 1.150$
   - Arm 2: $\frac{2}{2} + \frac{\sqrt{\log_{10} 7}}{\sqrt{2}} = 1.650$
   - Arm 3: $\frac{3}{2} + \frac{\sqrt{\log_{10} 7}}{\sqrt{2}} = 2.150$
   - Arm 4: $\frac{V}{1} + \frac{\sqrt{\log_{10} 7}}{\sqrt{1}} = V + 0.919$

   Since Arm 4 was picked as the 8th pull its UCB value must have been the largest, meaning its value was greater than Arm 3's value (since Arm 3's is otherwise the largest): $V + 0.919 > 2.150$, or in other words $V > 1.231$. In other words the smallest value it can have is 1.231.

   The analysis for what its largest value might have been comes from analyzing what value it might have had after 6 arm pulls were made – its value had to be lower than the value for whatever other arm might have been pulled. First, notice that N_4=1 means that it has only been pulled once, as part of the initialization of all arms, pulling each once. So the 7th arm pull had to

be either Arm 1, 2, or 3. I'll do the analysis for the case where arm 3 was the armed pulled 7th. The analyses for the other arms are analogous, but giving lower values for arm 4's value, and since we want the largest possible value it's arm 3 that gives it to us.

Let do a similar table of the values as we did before, but listing the values after 6 arm pulls, using W as the value for arm 3 at that point:

- Sum_1=1 , N_1=2
- Sum_2=2, N_2=2
- Sum_3=W, N_3=1
- Sum_4=V, N_4=1

Arm 4 has again only been pulled once at this point, so its value is still V. If arm 3 was pulled that means W>V. (I'm ignoring the case where W=V and arm 3 is pulled due to a tie breaker, because that gets us to the same number.) Let's use R to refer to the reward that arm 3 gave us for this 7th overall pull, its second pull of arm 3. Recall that the original problem states that its value after two pulls is 3, or in other words W+R=3, which I'll rewrite as R=3-W. But all arms always give positive rewards, so R>0, which means 3-W>0 or W<3. Since we also just analyzed that V<W, that means V<3. (The comparable analyses for the other two arms give V<2 for arm 2, and V<1 for arm 1. Thus we still might have gotten into the situation given at the start for values of V approaching 3, but for any such case where V>2 it must have been the case that the 7th arm pulled was arm 3. Note also that this analysis tells us that arm 1 could *not* have been the 6th arm pulled, because that requires V<1, but we already determined that V>1.231.)

So, to summarize, if you consider all the possible situations where arm 4 would have been pulled 8th, given the values we were given at the start, whatever value arm 4 had must have been between 1.231 and 3.

2. Consider a multi-armed bandit with two arms. Arm 1 always gives a reward of 1 on each pull. Arm 2 always gives a reward of 0 on each pull. Other than during the initialization when each arm is pulled once, would arm 2 ever get pulled again? If no, explain why. If yes, give a value for the number of pulls.

[The question says Arm 1 always gives a 1 and Arm 2 always gives a 0, but in the solution I accidentally swapped the two. Rather than changing the solution out from under people I'm instead leaving it, but adding this explanation that if switch all the Arm 1s and Arm 2s you get the correct answer.]

After the initializing first two pulls we get the following UCB values for the two arms:

- Arm 1: $\frac{0}{1} + \frac{\sqrt{\log_{10} 2}}{\sqrt{1}} = 0.549$
- Arm 2: $\frac{1}{1} + \frac{\sqrt{\log_{10} 2}}{\sqrt{1}} = 1.549$

So the third arm pull will be for arm 2, since its value is larger. Let's do this again to figure out the fourth pull.

- Arm 1: $\frac{0}{1}+\frac{\sqrt{\log_{10} 3}}{\sqrt{1}}=0.691$
- Arm 2: $\frac{2}{2}+\frac{\sqrt{\log_{10} 3}}{\sqrt{2}}=1.488$

If we keep doing this the values for Arm 1 will keep increasing, and the values for arm 2 will keep decreasing. If there have been a total of N pulls with 1 for arm 1 and N-1 for arm 2 the formulas would be:

- Arm 1: $\frac{0}{1}+\frac{\sqrt{\log_{10} N}}{\sqrt{1}}=\sqrt{\log_{10} N}$
- Arm 2: $\frac{N-1}{N-1}+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}=1+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}$

Based on these values arm 2 would keep getting pulled as long as $1+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}>\sqrt{\log_{10} N}$. The number we're seeking is the first value for which this is violated, namely $1+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}<\sqrt{\log_{10} N}$. The quick-and-dirty way to determine N is to just plug in values for N till you get to the right value. You could get in the ballpark by going up by increments of 10 for N, with whichever one is larger given in red:

N	$\sqrt{\log_{10} N}$	$1+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}$
10	1	1.333333
20	1.140627	1.261678
30	1.215369	1.225688
40	1.265725	1.202678

This shows that the value of N is between 30 and 40. If you were to then go up one-by-one from 30 you would get the following:

N	$\sqrt{\log_{10} N}$	$1+\frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}$
30	1.215369	1.225688
31	1.221213	1.222962
32	1.226846	1.220348

This means after the initial 2 pulls UCB would say to pull arm you would pull arm 2 for arm pulls 3-31, then finally pull arm 1 on arm pull 32.

3. True/False: Consider a multi-armed bandit with five arms. All five arms are identical and give a reward of 1 on each pull. After 10 arm pulls using the UCB algorithm each arm will have been pulled twice, regardless of how you break ties when arms have equal UCB values.

True. After the first 5 pulls the UCB values would be as follows:

- Arm 1: $\frac{1}{1}+\frac{\sqrt{\log_{10} 5}}{\sqrt{1}}=1.836$
- Arm 2: $\frac{1}{1}+\frac{\sqrt{\log_{10} 5}}{\sqrt{1}}=1.836$
- Arm 3: $\frac{1}{1}+\frac{\sqrt{\log_{10} 5}}{\sqrt{1}}=1.836$
- Arm 4: $\frac{1}{1}+\frac{\sqrt{\log_{10} 5}}{\sqrt{1}}=1.836$
- Arm 5: $\frac{1}{1}+\frac{\sqrt{\log_{10} 5}}{\sqrt{1}}=1.836$

They are all tied at the same value. The algorithm does not care how we break ties, it will give the same result for the sixth arm pull regardless of which of the arms we pull. Let's arbitrarily say arm 1 gets pulled. We wind up with the following:

- Arm 1: $\frac{2}{2}+\frac{\sqrt{\log_{10} 6}}{\sqrt{2}}=1.624$
- Arm 2: $\frac{1}{1}+\frac{\sqrt{\log_{10} 6}}{\sqrt{1}}=1.882$
- Arm 3: $\frac{1}{1}+\frac{\sqrt{\log_{10} 6}}{\sqrt{1}}=1.882$
- Arm 4: $\frac{1}{1}+\frac{\sqrt{\log_{10} 6}}{\sqrt{1}}=1.882$
- Arm 5: $\frac{1}{1}+\frac{\sqrt{\log_{10} 6}}{\sqrt{1}}=1.882$

We now have a tie between arms 2-5. If we arbitrarily choose to pull arm 2 from those 4 arms we get the following:

- Arm 1: $\frac{2}{2}+\frac{\sqrt{\log_{10} 7}}{\sqrt{2}}=1.650$
- Arm 2: $\frac{2}{2}+\frac{\sqrt{\log_{10} 7}}{\sqrt{2}}=1.650$
- Arm 3: $\frac{1}{1}+\frac{\sqrt{\log_{10} 7}}{\sqrt{1}}=1.919$
- Arm 4: $\frac{1}{1}+\frac{\sqrt{\log_{10} 7}}{\sqrt{1}}=1.919$
- Arm 5: $\frac{1}{1}+\frac{\sqrt{\log_{10} 7}}{\sqrt{1}}=1.919$

The same process continues – first with a tie between arms 3-5 (let's pick arm 3 to pull) and then with a tie between arms 4 and 5 (let's pick arm 4 to pull). After those two arm pulls we wind up with the following:

- Arm 1: $\frac{2}{2}+\frac{\sqrt{\log_{10} 9}}{\sqrt{2}}=1.691$
- Arm 2: $\frac{2}{2}+\frac{\sqrt{\log_{10} 9}}{\sqrt{2}}=1.691$
- Arm 3: $\frac{2}{2}+\frac{\sqrt{\log_{10} 9}}{\sqrt{2}}=1.691$
- Arm 4: $\frac{2}{2}+\frac{\sqrt{\log_{10} 9}}{\sqrt{2}}=1.691$
- Arm 5: $\frac{1}{1}+\frac{\sqrt{\log_{10} 9}}{\sqrt{1}}=1.977$

That represents a total of 9 pulls thus far. Using UCB to again pick among the arms results in choosing arm 5, since it has the largest UCB value. Choosing that as the 10th pull results in each arm being pulled 2 times.

4. Consider a multi-armed bandit with two arms. After the first pull of each arm Arm 1 gave 0 and Arm 2 gave 1. How many additional arm pulls would it take to pull Arm 1 again, assuming in the meantime Arm 2 always gave a 0 for each further pull?

   After the initializing first two pulls we get the following UCB values for the two arms:

   - Arm 1: $\frac{0}{1} + \frac{\sqrt{\log_{10} 2}}{\sqrt{1}} = 0.549$
   - Arm 2: $\frac{1}{1} + \frac{\sqrt{\log_{10} 2}}{\sqrt{1}} = 1.549$

   Arm 2 looks better, so we would pull it next, resulting in:

   - Arm 1: $\frac{0}{1} + \frac{\sqrt{\log_{10} 3}}{\sqrt{1}} = 0.691$
   - Arm 2: $\frac{1}{2} + \frac{\sqrt{\log_{10} 3}}{\sqrt{2}} = 0.988$

   If we keep pulling arm 2 after N total pulls we get:

   - Arm 1: $\frac{0}{1} + \frac{\sqrt{\log_{10} N}}{\sqrt{1}} = \sqrt{\log_{10} N}$
   - Arm 2: $\frac{1}{N-1} + \frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}$

   We're asking for the lowest value of N for which the first of these is larger than the second, because that would be when we would finally pull arm 1. In other words we want:

   $$\sqrt{\log_{10} N} > \frac{1}{N-1} + \frac{\sqrt{\log_{10} N}}{\sqrt{N-1}}$$

   When N=5 we get 0.836 > 0.668. In other words we would pull arm 2 an additional 3 times after its first pull, for a total of 4 pulls, and arm 1 would still have just its initial pull. At that point arm 1 would have the larger UCB value and would be pulled.