CS7180: Special Topic - Artificial Intelligence (Spring 2022) — Christopher Amato
Northeastern University — Due Feb 2, 2022

Exercise 1: Multi-armed Bandits

Please remember the following policies:

- Exercise due at **11:59 PM EST Feb 2, 2022**.
- Submissions should be made electronically on Canvas. Please ensure that your solutions for both the written and programming parts are present. You can upload multiple files in a single submission, or you can zip them into a single file. You can make as many submissions as you wish, but only the latest one will be considered.
- For **Written** questions, solutions may be handwritten or typeset. If you write your answers by hand and submit images/scans of them, please please ensure legibility and order them correctly in a single PDF file.
- The PDF file should also include the figures from the **Plot** questions.
- For both **Plot** and **Code** questions, submit your source code along with reasonable documentation.
- You are welcome to discuss these problems with other students in the class, but you must understand and write up the solution and code yourself. Also, you *must* list the names of all those (if any) with whom you discussed your answers at the top of your PDF solutions page.
- Each exercise may be handed in up to two days late (24-hour period), penalized by 10% per day late. Submissions later than two days will not be accepted.
- Contact the teaching staff if there are medical or other extenuating circumstances that we should be aware of.

1. **1 point.** (RL2e 2.2) *Exploration vs. exploitation.*
   Written: Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ε-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ε case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

2. **1 point.** (RL2e 2.4) *Varying step-size weights.*
   Written: If the step-size parameters, α_n, are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by Equation 2.6. What is the weighting on each prior reward for the general case, analogous to Equation 2.6, in terms of the sequence of step-size parameters?

3. **2 points.** *Bias in Q-value estimates.*
   Written: Recall that $Q_n \triangleq \frac{R_1+\ldots+R_{n-1}}{n-1}$ is an estimate of the true expected reward q_* of an arbitrary arm a. We say that an estimate is *biased* if the expected value of the estimate does not match the true value, i.e., $\mathbb{E}[Q_n] \neq q_*$ (otherwise, it is *unbiased*).

   (a) Consider the *sample-average* estimate in Equation 2.1. Is it biased or unbiased? Explain briefly.

   For the remainder of the question, consider the *exponential recency-weighted average* estimate in Equation 2.5. Assume that $0 < \alpha < 1$ (i.e., it is strictly less than 1).

   (b) If $Q_1 = 0$, is Q_n (for $n > 1$) biased? Explain briefly.
   (c) Derive condition(s) for Q_1 for when Q_n will be unbiased.
   (d) Show that Q_n is an unbiased estimator as $n \to \infty$ (which is often referred to as*asymptotically unbiased*).
   (e) Why should we expect that the *exponential recency-weighted average* will be biased in practice? Think about what happens to Q_1 or α in practice.

4. **1 point.** (RL2e 2.9) *Gradient Bandit*
   Written: Show that in the case of two actions, the soft-max distribution is the same as that given by the logistic, or sigmoid, function often used in statistics and artificial neural networks.

For the remainder of this assignment, you will be implementing the 10-armed testbed described in Section 2.3, reproducing some textbook figures based on this testbed, and performing further experimentation on bandit algorithms. Some general tips:

- Some of the experiments might take a while. During development, you may want to use smaller numbers of steps and trials to make iteration much faster. Here are some possible settings:
  - Tiny: 100 steps, 20 trials – for fast iteration, should take < 1 second
  - Small: 100 steps, 200 trials – for development and debugging purposes, should take < 10 seconds
  - Medium: 1000 steps, 2000 trials – the final setting in this assignment and the setting used in the textbook, should take < 5 minutes
- The above times were based on our implementation running on a single CPU thread of a Lenovo ThinkPad T480s (circa 2018) laptop. Optionally, you might consider multiprocessing to improve speed. You may also consider adjusting the settings above of steps/trials in order to achieve fast iteration.

5. **1 point.** *Implementing the* 10*-armed testbed for further experimentation in the remainder of the assignment.*
   Code: Implement the 10-armed testbed described in the first paragraph Section 2.3 (p. 28).
   Read the description carefully.
   Plot: To test that your testbed is working properly, produce a plot similar in style to Figure 2.1 by pulling each arm many times and plotting the distribution of sampled rewards. You can use any type of plot that makes this point effective, e.g., a violin plot, or a scatterplot with some jitter in the horizontal axis to show the sample density more effectively.

6. **2 points.** *Reproducing Figure 2.2.*
   Code: Implement the ε-greedy algorithm with incremental updates. Note that in the graph: "All the methods formed their action-value estimates using the sample-average technique (with an initial estimate of 0)."
   Plot: Reproduce both plots shown in Figure 2.2, with the following modifications:
   - Use 10^3 steps with 2000 independent runs (same as in text). Make sure that all methods are evaluated on the same set of 2000 10-armed bandit problems.
   - For the reward plot, add an extra constant upper bound line corresponding to the best possible *average* performance in your trials, based on the known true expected rewards $q_*(a)$. That is, the line should correspond to $\max_a q_*(a)$, averaged over all trials. (Why is this the appropriate upper bound?)
   - For each curve (including the upper bound), also plot confidence bands corresponding to (1.96$\times$ standard error) of your rewards.
     The standard error of the mean is defined as the standard deviation divided by $\sqrt{n}$: $\frac{\sigma}{\sqrt{n}}$
     This corresponds to a $\sim 95\%$ confidence interval around the average performance. In other words, our uncertainty in the average performance decreases as the number of trials increases. See the following for an example of plotting a confidence band in `matplotlib.pyplot`: `https://matplotlib.org/3.3.1/gallery/lines_bars_and_markers/fill_between_demo.html#example-confidence-bands`

   Written: Do the averages reach the asymptotic levels predicted in class?

7. **2 points.** *Reproducing and supplementing Figures 2.3 and 2.4.*
Code: Implement the ε-greedy algorithm with optimistic initial values, and the bandit algorithm with UCB action selection.
Plot: Reproduce the plots shown in both Figures 2.3 and 2.4, with the following modifications:

   - Use 10^3 steps with 2000 independent runs (same as in text). Make sure that all methods are evaluated on the same set of 2000 10-armed bandit problems.
   - We will merge the results for both Figures 2.3 and 2.4 and plot additional curves. Figure 2.3 should show % optimal action and Figure 2.4 should show the average reward. Both plots should each contain 5 curves:
     - ε-greedy ($Q_1 = 0, \varepsilon = 0$)
     - ε-greedy ($Q_1 = 5, \varepsilon = 0$)
     - ε-greedy ($Q_1 = 0, \varepsilon = 0.1$)
     - ε-greedy ($Q_1 = 5, \varepsilon = 0.1$)
     - UCB ($c = 2$)
   - For each curve (including the upper bound), also plot confidence bands corresponding to ($1.96\times$ standard error) of your rewards.
   - For the reward plot, add an extra constant upper bound line corresponding to the best possible *average* performance in your trials, based on the known true expected rewards $q_*(a)$. That is, the line should correspond to $\max_a q_*(a)$, averaged over all trials. (Why is this the appropriate upper bound?)

   Written: We have seen the spike for optimistic initialization in class. Observe that UCB also produce spikes in the very beginning. Explain in your own words why the spikes appear (both the sharp increase and sharp decrease). Analyze and use your experimental data as further empirical evidence to back your reasoning.