Multi-Agent Systems

Homework Assignment 4

Martynas Vaznonis (2701013)
m.vaznonis@student.vu.nl

4 Fictitious Play

1. The results are player 1 plays the actions A, B, C with probabilities 0, 0.5, 0.5 respectively and player 2 plays the actions W, X, Y, Z with probabilities 0, 0.6, 0.4, 0 respectively (see `fictitious_play.py`).

2. The results do make sense as can be seen by examining the reward table. Both agents make each other indifferent as to what actions to take. Player 2 plays X with $x = \frac{3}{5}$ and Y with $y = \frac{2}{5}$, thus the expected utility for player 1 is $\mathbb{E}u_1(B, s_2^*) = \mathbb{E}u_1(C, s_2^*) = 3.8$. Player 1 plays B with $b = \frac{1}{2}$ and C with $c = \frac{1}{2}$, thus the expected utility for player 2 is $\mathbb{E}u_2(X, s_1^*) = \mathbb{E}u_1(Y, s_1^*) = 3.5$. The other actions have lower expected utilities and so all have a probability of 0 and are not mixed.

5 Monte Carlo simulation

5.2 Warming up

1. The estimated mean value is $\mathbb{E}(\cos^2(X)) \approx 0.567$ with variance $\sigma^2 \approx 0.12$. The 99.9% confidence interval is 0.567 ± 0.011. See `monte_carlo.py`.

5.3 Quantifying the significance of an observed correlation

2. Here $S = \varphi(A)$. Although because the final performance of neural networks is stochastic, more accurately $S \sim \varphi(A)$. To test whether the positive correlation is significant the fraction of A for which $A > A'$ but $\varphi(A) \leq \varphi(A')$ can be computed, $p = \frac{|\{(A,A')|\forall A,A':A\neq A' \wedge A>A' \wedge \varphi(A)\leq\varphi(A')\}|}{|\{(A,A')|\forall A,A':A\neq A'\}|}$. If only a small proportion (p) of all possible pairs violate the correlation, then it is significant with probability $1 - p$.

6 Exploration versus Exploitation: Thompson Sampling for Multi-Armed Bandits

6.2 Thompson's Bayesian update rule

Questions:

1. Figure 1 shows the estimate of p as a function of the number of sampled points. It can be easily seen that the right probability is quickly reached and that uncertainty tapers with subsequent sampling. For implementation see `thompson.py`.

6.3 Thompson sampling for K-armed bandit Problem

Questions:

2. See `thompson.py`.

3. Figure 2 shows a comparison between Thompson sampling and UCB. Regret is defined as the $r_i = \max_k(p_k) - p_i$ for each arm i. As can be seen Thomson sampling outperforms UCB in 2 out of 4 experiments and can thus be considered competitive. Furthermore, from my experimentation, the better values of c were highly dependent on the values p_k. Thus, every instance of the K-bandit problem would require tuning this value. On the other hand, the Thompson sampling method always performed well, even if it was sometimes outperformed by a well-tuned UCB instance.

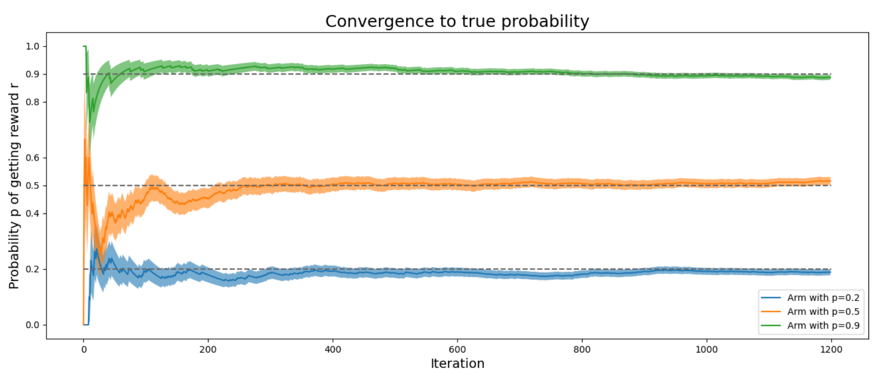

Figure 1: The mean estimate and variance of probability p. The shaded area is the standard deviation instead of variance to make the tapering uncertainty effect more salient.

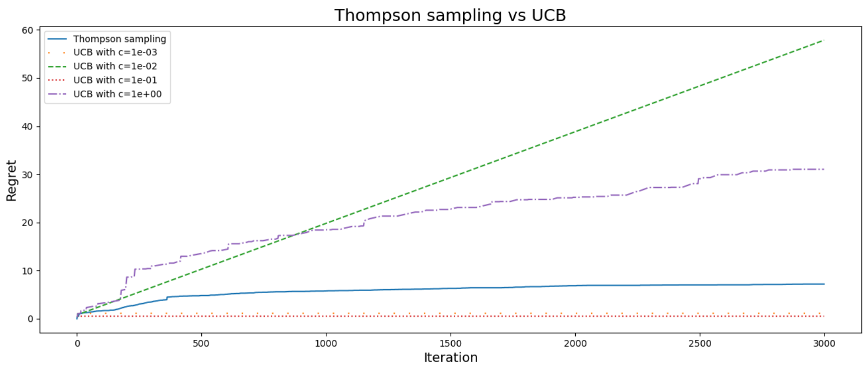

Figure 2: Thompson sampling (solid line) versus UCB (dashed and doted lines) with respect to the cumulative regret. UCB outperforms Thompson sampling with a well tuned c parameter, but grossly underperforms if the values are ill-chosen. The values of p used in this plot are 0.462, 0.956, and 0.975 for arms 1, 2, and 3 respectively.