10 Multi-Armed Bandit

Exercise 10.1

Tell if the following statements about MAB are true or false and provide motivations for your choice.

1. The MDP corresponding to a MAB setting has no state.
2. The MDP corresponding to a MAB setting has no transition probabilities.
3. An algorithm which always uses the greedy choice (maximize the empirical expected reward) might get stuck to suboptimal solutions.
4. We are able to solve the exploration/exploitation problem by considering either upper or lower bounds on the expected rewards.
5. In a frequentist framework, if we base our sequential policy on a MAB by considering only $\hat{R}(a_i)$ we are not using all the information we have about the estimates.
6. Uncertainty over quantities is usually handled by explorative choices in the MAB setting.

Exercise 10.2

Tell if the following problems can be modeled as MAB and explain why.

1. Maze escape;
2. Pricing goods with stock;
3. Pricing goods without stock;
4. Optimal bandwidth allocation (send the largest amount of information without congesting the band);
5. Web ads placement;
6. Weather prediction (with multiple experts).

If they fit the MAB setting, is the environment adversarial or stochastic?

Exercise 10.3

State if the following are applicable to generic RL problems or MAB problems. Motivate your answers.

1. We should take into account state-action pairs one by one to estimate the value function;
2. Past actions you perform on the MDP might influence the future rewards you will gain;
3. The Markov property holds;
4. The time horizon of the episode is finite.

Exercise 10.4

The ε-greedy algorithm selects the best action except for small percentages of times $\varepsilon \in (0, 1)$, where all the actions are considered. Consider a MAB stochastic setting.

1. Is this algorithm converging to the optimal strategy (in some sense)?
2. If not, propose a scheme which has the chance of converging to the optimal solution.
3. Are we in a MAB perspective if we are using this algorithm?

Exercise 10.5

Write the formula for the minimum regret we might have on average over $T = \lceil e^{10} \rceil$ time steps in the case we have a stochastic MAB with 3 arms and expected rewards:

$$R(a_1) = 0.2 \tag{10.1}$$
$$R(a_2) = 0.4 \tag{10.2}$$
$$R(a_3) = 0.7 \tag{10.3}$$

and each distribution $\mathcal{R}(a_i)$ is Bernoulli.

Note that the KL divergence for Bernoulli variables with means p and q is:

$$KL(p, q) = p \log\left(\frac{p}{q}\right) + (1 - p) \log\left(\frac{(1 - p)}{(1 - q)}\right).$$

Hints: $\log(\frac{0.2}{0.7}) = -1.25$, $\log(\frac{0.8}{0.3}) = 0.98$, $\log(\frac{0.4}{0.7}) = -0.56$ and $\log(\frac{0.6}{0.3}) = 0.69$.

Is it possible that your algorithm achieves lower regret? If so, provide an example.

Exercise 10.6

Provide examples of either Bayesian or frequentist MAB algorithm showing the following properties:

1. It incorporates expert knowledge about the problem in the arms;
2. It provides tight theoretical lower bound on the expected regret in the stochastic setting;
3. It provides tight theoretical upper bound on the expected regret in the stochastic setting;
4. At each turn, it modifies only the statistics of the chosen arm.

Motivate your answers.

Exercise 10.7

Consider the following Höeffding bounds (supposed to hold with probability at least δ) for a MAB stochastic setting:

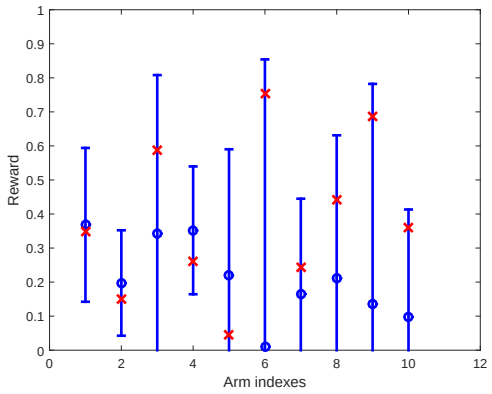

Figure 10.1: Bounds in a MAB with $N = 10$.

where the blue bars are the Höeffding bounds for the expected rewards, the blue circles are the estimated expected rewards, and the red crosses are the real expected rewards.

1. Which arm would a UCB1 algorithm choose for the next round?
2. Do you think that Figure 10.1 might be the results obtained by running UCB1 for several rounds?

3. Which arm will UCB1 converge to if $T \to \infty$?
4. Which arm is the one which we pulled the most so far?

Motivate your answers.

Exercise 10.8

Consider the Thompson Sampling algorithm. Assume to have the following posterior distributions $Beta_i(\alpha_t, \beta_t)$ for arms $\mathcal{A} = \{a_1, \ldots, a_5\}$ rewards, which are distributed as Bernoulli r.v.:

$$
\begin{aligned}
a_1 &: \quad \alpha_t = 1 \quad \beta_t = 5 \\
a_2 &: \quad \alpha_t = 6 \quad \beta_t = 4 \\
a_3 &: \quad \alpha_t = 11 \quad \beta_t = 23 \\
a_4 &: \quad \alpha_t = 12 \quad \beta_t = 33 \\
a_5 &: \quad \alpha_t = 28 \quad \beta_t = 21
\end{aligned}
$$

From these distribution you extract the following samples for the current round:

$$
\begin{aligned}
\hat{r}(a_1) &= 0.63 \\
\hat{r}(a_2) &= 0.35 \\
\hat{r}(a_3) &= 0.16 \\
\hat{r}(a_4) &= 0.22 \\
\hat{r}(a_5) &= 0.7
\end{aligned}
$$

1. Which arm would the TS algorithm play for the next round?
2. What considerations can be done in the case the real distributions of the arm rewards are Gaussian. Is the standard Bernoulli TS algorithm a viable option?
3. Assume we started the TS algorithm with uniform $Beta(1, 1)$ priors. What would UCB1 have chosen in the case of Bernoulli rewards for the next round?

Exercise 10.9

Tell whether the following problems can be modeled as MAB or an MDP (or none). Explain why.

1. Web ads display;
2. Play “Space Invaders”;

3. Weather prediction (with multiple experts);
4. Maze escape.

Exercise 10.10

Consider the UCB1 and the Thompson Sampling algorithms. Tell if the following statements are true for the two aforementioned algorithms and motivate your answers.

1. It requires the knowledge of a pair of conjugate prior/posterior distributions;
2. It manages automatically the exploration/exploitation tradeoff;
3. It relies on the "optimism in the face of uncertainty" paradigm;
4. It is able to incorporate a priori knowledge about the problem.

Exercise 10.11

Consider the Thompson Sampling algorithm. Assume to have the following posterior distributions $Beta_i(\alpha_t, \beta_t)$ for arms $\mathcal{A} = \{a_1, \ldots, a_5\}$ rewards, which are distributed as Bernoulli r.v.:

$$
\begin{array}{llll}
a_1\text{:} & \alpha_t = 1 & \beta_t = 6 & \hat{r}(a_1) = 0.13 \\
a_2\text{:} & \alpha_t = 4 & \beta_t = 6 & \hat{r}(a_2) = 0.55 \\
a_3\text{:} & \alpha_t = 31 & \beta_t = 31 & \hat{r}(a_3) = 0.4 \\
a_4\text{:} & \alpha_t = 8 & \beta_t = 1 & \hat{r}(a_4) = 0.90 \\
a_5\text{:} & \alpha_t = 5 & \beta_t = 7 & \hat{r}(a_5) = 0.95
\end{array}
$$

where $\hat{r}(a_i)$ are random samples extracted from the posterior distributions.

1. Which arm would play the TS algorithm in the next round?
2. Do you think that there is an arm that is more promising to be the best one?
3. What is the UCB1 bound for arms a_3 and a_5? Assume that the Bayesian setting started from uniform $Beta(1, 1)$ priors.

Exercise 10.12

For each one of the following statements, tell if it is true for the UCB1 and/or TS algorithms.

1. It relies on the assumption to know the family of the arms reward distributions (e.g., Bernoulli);

2. At each round, it modifies the statistics of all the arms;
3. It incorporates knowledge about the arms;
4. It is a randomized algorithm.

Exercise 10.13

Consider the Thompson Sampling algorithm. Assume to have the following posterior distributions $Beta_i(\alpha_t, \beta_t)$ for arms $\mathcal{A} = \{a_1, \ldots, a_5\}$ rewards, which are distributed as Bernoulli random variables with mean μ_i, and you extracted from them the samples $\hat{r}(a_i)$:

$$
\begin{array}{llllll}
a_1\text{:} & \alpha_t = 1 & \beta_t = 5 & \hat{r}(a_1) = 0.63 & \mu_1 = 0.1 \\
a_2\text{:} & \alpha_t = 6 & \beta_t = 4 & \hat{r}(a_2) = 0.35 & \mu_2 = 0.5 \\
a_3\text{:} & \alpha_t = 11 & \beta_t = 23 & \hat{r}(a_3) = 0.16 & \mu_3 = 0.3 \\
a_4\text{:} & \alpha_t = 12 & \beta_t = 25 & \hat{r}(a_4) = 0.22 & \mu_4 = 0.2 \\
a_5\text{:} & \alpha_t = 38 & \beta_t = 21 & \hat{r}(a_5) = 0.7 & \mu_5 = 0.6
\end{array}
$$

1. How much pseudo-regret the TS algorithm accumulated so far, assuming we started from uniform $Beta(1, 1)$ priors?
2. Which one of the arm reward posteriors is the most peaked one?
3. What would UCB1 have chosen for the next round? Assume Bernoulli rewards and that in the Bayesian setting we started from uniform $Beta(1, 1)$ priors?

Exercise 10.14

Tell if the following statements about Multi-Armed Bandit are true or false and provide motivations for your choice.

1. The MDP corresponding to a MAB setting has no actions.
2. One of the Bayesian approaches to solve the MAB setting resorts to a sampling procedure from posterior distributions.
3. The use of an ε-greedy algorithm to solve the MAB setting is a viable solution and provides an expected pseudo-regret of order $O(\log(\log(T)))$, where T is the number of rounds.

Exercise 10.15

Tell if the following statements about the Thompson Sampling (TS) algorithm are true or false. Motivate your answers.

1. The more an arm has been pulled, the more the posterior distribution of its reward is peaked;
2. The TS algorithm uses specific information about the reward being Bernoulli distributed;
3. The TS algorithm is a deterministic algorithm.

Exercise 10.16

Tell if the following statements about MAB are true or false. Motivate your answers.

1. In a setting with Gaussian rewards we can use the UCB1 algorithm to have sublinear regret.
2. An arm with a large upper confidence bound is likely to turn out to be the optimal one at the end of the MAB time horizon.
3. A MAB algorithm can be used in an MDP as long as the reward corresponding to the actions have the same distribution in every state.
4. The Thompson Sampling algorithm can be applied only if we have bernoulli rewards (success/failures) as rewards.
5. The design of MAB algorithm different from UCB1 might provide an expected pseudo-regret of order log(log(T)), T being the time horizon.
6. The upper bound on the regret of UCB1 scales linearly with the number of arms.
7. There exist situations in which we are able to obtain a regret lower than the one prescribed by the lower bound for MAB.
8. It is generally a viable option to use an epsilon-greedy exploration strategy on a MAB setting.
9. The selection of the arm provided by TS is stochastic.

Exercise 10.17

Tell if the following statements about MAB are true or false. Motivate your answers.

1. In a setting with Gaussian rewards we can use the TS algorithm to have sublinear regret.
2. The selection of the arm provided by UCB1 is deterministic.
3. In the presence of prior information on the arms rewards, we should use the UCB1

algorithm.

4. The upper bound on the regret of UCB1 and TS scales logarithmically with the time horizon.

5. The design of MAB algorithms different from TS might provides an expected pseudo-regret of order log(log(T)), T being the time horizon.

Exercise 10.18

Consider the Thompson Sampling algorithm applied to an environment with Bernoulli rewards, in which the prior is a Beta distribution initialized as a uniform one: $Beta(1, 1)$.
For each arm $a_i \in A = \{a_0, \ldots, a_4\}$, you are provided its posterior distributions $Beta_i(alpha_t; beta_t)$, the true reward mean mu_i, and the last sample extracted from the Beta $r(a_i)$:

a_0	$\alpha_t = 8,$	$\beta_t = 6,$	$r_t = 0.46,$	$mu_0 = 0.6$
a_1	$\alpha_t = 8,$	$\beta_t = 18,$	$r_t = 0.34,$	$mu_1 = 0.3$
a_2	$\alpha_t = 6,$	$\beta_t = 9,$	$r_t = 0.33,$	$mu_2 = 0.4$
a_3	$\alpha_t = 18,$	$\beta_t = 19,$	$r_t = 0.4,$	$mu_3 = 0.5$
a_4	$\alpha_t = 4,$	$\beta_t = 14,$	$r_t = 0.2,$	$mu_4 = 0.2$

1. How much pseudo-regret did the TS algorithm accumulate so far?

2. What would UCB1 have chosen for the next round?

3. Suppose the next reward can be chosen by an adversary, but keeping the same domain: how could this adversary behave if it knows that the agent is using TS? What about UCB1 instead? (no computations are required)

Exercise 10.19

Consider a MAB algorithm choosing the arm a_t and providing the following rewards R_t over a time horizon of $T = 10$ rounds.

$R_{1,t}$	**0**	1	1	0	**0**	1	0	1	**0**	0
$R_{2,t}$	1	**0**	1	1	1	1	**1**	1	1	**1**
$R_{3,t}$	0	1	**0**	**0**	0	**0**	1	**0**	1	1
a_t	1	2	3	3	1	3	2	3	1	2

(only the reward for the chosen arm is revealed to the algorithm) Moreover, assume that the expected value for the three arms' reward are $\mu_1 = 0.3$, $\mu_2 = 0.8$, and $\mu_3 = 0.6$.

1. Compute the cumulated reward, the regret and the pseudo-regret for the algorithm over the time horizon T. (Recall that the regret is computed over realizations, while the pseudo-regret is computed over expected values);

2. Compute the values for the UCB1 bounds for at time $t = 5$ for the three arms. Do you think that the algorithm used in this setting can be the UCB1?

3. Do you think that the algorithm used in this setting might be the Thompson sampling?

Exercise 10.20

Consider the following snippet of code:

```
for t in range(1, T+1):
pulled = np.argwhere(criterion == criterion.max()).reshape(-1)
reward = rew(pulled)

n_pulls[pulled] = n_pulls[pulled] + 1
exp_payoffs[pulled] = ((exp_payoffs[pulled] *
(n_pulls[pulled] - 1.0) + reward) / n_pulls[pulled])
for k in range(0, n_options)
criterion[k] = exp_payoffs[k] + np.sqrt(2 * t / n_pulls[k])
```

1. Describe the procedure (what algorithm it is implementing) and the purpose (which kind of problem it is solving) of the above snippet of code.

2. Is it correct? In the case the algorithm is not correct, propose a modification to fix the problem. In the case the algorithm is correct, state the theoretical guarantees of such an algorithm.

3. Are there other available methods to solve this problem? Are they requiring some specific assumptions to be applied to the same setting the algorithm in the snippet is used?

Exercise 10.21

Assume to have a stochastic Multi-Armed Bandit (MAB) with 3 arms with average reward:

$$R(a_1) = 0.1,$$
$$R(a_2) = 0.6,$$
$$R(a_3) = 0.3,$$

and each distribution $\mathcal{R}(a_i)$ is a Bernoulli.

1. Write the asymptotic minimum expected pasudo-regret we might have on average over $T = e^{10}$ time steps.

2. If we apply the UCB1 algorithm, which is the upper bound on the expected pseudo-regret? Is it larger or smaller than the previous one?

3. What can we tell about the minimum regret we might have on the above problem? And about the minimum regret of UCB1?

Note that the KL divergence for Bernoulli variables with means p and q is:

$$KL(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{(1-p)}{(1-q)}\right)$$

and (if necessary) that $\log(\frac{0.1}{0.6}) = -1.8$, $\log(\frac{0.9}{0.4}) = 0.8$, $\log(\frac{0.3}{0.6}) = -0.7$, $\log(\frac{0.7}{0.4}) = 0.6$, $\log(\frac{0.1}{0.3}) = -1.1$, and $\log(\frac{0.9}{0.7}) = 0.25$.

Solutions

Answer of exercise 10.1

1. FALSE It has a single state.
2. FALSE The transition probability matrix is $P(s, a_i|s) = 1$.
3. TRUE There are cases in which if we only consider the expected values of the rewards as decision tool, we might discard good options even if we only gathered small evidence from them.
4. FALSE Pessimistic choices would not solve the exploration/exploitation dilemma since, this way, we do not provide incentives to explore the options which are considered suboptimal so far.
5. TRUE We are also allowed to consider concentration inequalities which provides us information about the uncertainty we have about the reward estimate.
6. TRUE Sometimes, we want to reduce uncertainty by picking suboptimal choices which allow us to gather more information that will be beneficial in the future.

Answer of exercise 10.2

1. NO There are plenty of states in a maze escape problem (the reward cannot be modeled as a single distribution).
2. NO We are limited in the number of successes, thus we might stop the problem at some point (it can be a generalized MAB, in the specific a budget MAB).
3. YES Rewards are stochastic if we consider a distribution of users or adversarial if we assume to have a single malicious buyer.
4. YES If we consider the traffic in the bandwidth as a stochastic event.
5. YES Assuming, for instance, a pay-per-click scheme and a click event which is a stochastic event.
6. NO Each time we predict we have rewards coming from all the arms (expert problem). If we use only the feedback coming from the pulled arm, we can use the MAB model, even if we would discard some useful information by doing this.

Answer of exercise 10.3

Answer of exercise 10.4

1. No, it can not reach the optimal policy since it will always select the suboptimal arms with a fixed probability. This leads to a regret of the order $\varepsilon T \gg O(\log T)$.
2. If we use a strategy s.t. $\lim_{t\to\infty} \varepsilon(t) = 0$ we might converge.
3. Even if this strategy converges, you do not have any assurance about the regret we are suffering in the process of learning. There exists a theoretical analysis of a version of the ε-greedy algorithm in:
   Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multi-armed bandit problem." Machine learning 47.2-3 (2002): 235–256
   The solution proposed can be applied only by knowing the gaps Δ_i between arms, otherwise the bound might not hold.

Answer of exercise 10.5

If we assume that T is sufficiently large, we have:

$$R_T \geq \log T \sum_{a_i \neq a^*} \frac{\Delta_i}{KL(\mathcal{R}(a_i), \mathcal{R}(a^*))},$$

thus in our case:

$$\begin{aligned} R_T &\geq \log e^{10} \left(\frac{0.7 - 0.2}{KL(0.2, 0.7)} + \frac{0.7 - 0.4}{KL(0.4, 0.7)} \right) \\ &= 10 \left(\frac{0.5}{0.2(-1.25) + 0.8(0.98)} + \frac{0.3}{0.4(-0.56) + 0.6(0.69)} \right) \\ &= 25.1528 \end{aligned}$$

We might violate our lower bound in some cases, e.g.:

- if we consider a subset of MAB settings;
- if we consider a single realization of the rewards.

Answer of exercise 10.6

1. TS As usual we can use prior distributions to incorporate information about the problem.
2. NONE The lower bound is defined over the problem and not by basing on the algorithm we used.

3. UCB1 and TS Provide order optimal theoretical upper bound on the expected regret in the stochastic setting. TS matches also the constant (KL-UCB has the same assurance).

4. TS It only modifies the Beta distribution corresponding to the pulled arm.

Answer of exercise 10.7

1. a_6 since it is the one with the highest upper bound.

2. No, since in principle UCB1 keeps all the upper bounds at similar level.

3. a_6, since it is the one with highest expected value.

4. a_2, since it is the one with smallest bounds (which are inversely proportional to the number of pulls).

Answer of exercise 10.8

1. Thompson sampling will choose the arm a_i with the largest sampled $\hat{r}(a_i)$, which in this case is a_5.

2. In the case we are considering Gaussian rewards it is not meaningful to use Beta distribution as prior. For instance, if the reward are Gaussian they might provide also negative values, while the support of the Beta is $[0, 1]$. Thus, it makes no sense to instantiate a problem with Beta prior when you have Gaussian rewards.

3. Since we started with uniform prior and at each round we collected a success or a failure we are currently at round:

$$t = 4 + 8 + 32 + 43 + 47 = 134,$$

while the UCB1 upper bound $U_t(a_i)$ is of the form:

$$U_t(a_i) = \frac{\alpha_t - 1}{\alpha_t + \beta_t - 2} + \sqrt{\frac{2 \log t}{\alpha_t + \beta_t - 2}},$$

thus:

$$U_t(a_1) = \frac{0}{4} + \sqrt{\frac{2\log 134}{4}}$$
$$U_t(a_2) = \frac{5}{8} + \sqrt{\frac{2\log 134}{8}}$$
$$U_t(a_3) = \frac{10}{32} + \sqrt{\frac{2\log 134}{32}}$$
$$U_t(a_4) = \frac{11}{43} + \sqrt{\frac{2\log 134}{43}}$$
$$U_t(a_5) = \frac{27}{47} + \sqrt{\frac{2\log 134}{47}}$$

In this case the UCB1 algorithm chooses the arm a_i s.t. $i = \arg\max_i U_t(a_i)$, in this case a_2.

Answer of exercise 10.9

1. MAB, stochastic: we can select a single ad at each time point and receive a feedback (click/no-click) from a user, sampled from a user distribution.
2. MDP: the state is not unique and the optimal action is dependent on the game state.
3. Expert: we receive feedback from all the considered experts. If we consider a feedback coming only from the chosen expert it is a MAB.
4. MDP: the optimal action we might perform in a maze is dependent on the position we are in.

Answer of exercise 10.10

1. TS: it relies on a Bayesian framework, it requires to have a prior and a way of updating this prior. One of the most common way is to use conjugate prior/posterior distributions.
2. TS and UCB1: all the MAB algorithm are specifically designed to minimize the regret and thus they manage automatically the exploration/exploitation tradeoff.
3. UCB1: it is based on the computation of the Höeffding upper confidence bounds, which provides an optimistic estimates of the expected reward of the arms.
4. TS: since it is a Bayesian method, it can be naturally extended to include information about the problem by the modification of the prior.

Answer of exercise 10.11

1. Arm a_5 since its sample is the highest one.
2. Arm a_4 only provided positive samples, thus, on average, it is the most promising one.
3. Since we have an overall number of pulls of $t = 5 + 8 + 60 + 7 + 10 = 90$ we have:

$$u(a_3) = \frac{30}{60} + \sqrt{\frac{2\log(90)}{60}}$$

$$u(a_5) = \frac{4}{10} + \sqrt{\frac{2\log(90)}{10}}$$

Answer of exercise 10.12

1. TS: we need to know a conjugate pair prior-posterior specific for the distribution of the rewards before executing the method.
2. UCB1: it updates the bound of all the arms, since in its numerator we have $logt$, t being the current round.
3. TS: by choosing the appropriate prior we are able to introduce some a priori information about the process in the prior.
4. TS: it extracts a sample from each posterior distribution of the arms at each turn.

Answer of exercise 10.13

1. The algorithm pulled the arms $T_{i,t} = \alpha_t + \beta_t - 2$ times, i.e., $T_{1,t} = 4$, $T_{2,t} = 8$, $T_{3,t} = 32$, $T_{4,t} = 35$, and $T_{5,t} = 57$. The overall number of pulls is $t = \sum_i T_{i,t} = 136$. The pseudo-regret can be computed as:

$$\begin{aligned} R_T(\mathcal{U}) &= t\ \mu_5 - \sum_{i \neq 5} T_{i,t}\mu_i \\ &= 136 \cdot 0.6 - (4 \cdot 0.1 + 8 \cdot 0.5 + 32 \cdot 0.3 + 57 \cdot 0.2) = 56.2. \end{aligned}$$

2. Since it is the one which is the most pulled one, a_5.
3. The empirical expected reward for the different arms is $\hat{\mu}_{i,t} = \frac{\alpha_t - 1}{T_{i,t}}$ and the bound

is computed as $b_{i,t} = \sqrt{\frac{2\ln(t)}{T_{i,t}}}$, therefore:

$$B_{1,t} := \hat{\mu}_{1,t} + b_{1,t} = \frac{0}{4} + \sqrt{\frac{2\ln 136}{4}};$$
$$B_{2,t} := \hat{\mu}_{2,t} + b_{2,t} = \frac{5}{8} + \sqrt{\frac{2\ln 136}{8}};$$
$$B_{3,t} := \hat{\mu}_{3,t} + b_{3,t} = \frac{10}{32} + \sqrt{\frac{2\ln 136}{32}};$$
$$B_{4,t} := \hat{\mu}_{4,t} + b_{4,t} = \frac{11}{35} + \sqrt{\frac{2\ln 136}{35}};$$
$$B_{5,t} := \hat{\mu}_{5,t} + b_{5,t} = \frac{37}{57} + \sqrt{\frac{2\ln 136}{57}};$$

The algorithm would have pulled the arm with the largest value for $B_{i,t}$.

Answer of exercise 10.14

1. FALSE: determining the correct action among a finite set is the goal of MAB algorithms.
2. TRUE: it is Thompson Sampling, which selects the arm to pull depending on the largest sample generated from posterior distributions.
3. FALSE: The stochastic MAB problem has a lower bound of $O(\log(T))$, therefore, no algorithm can achieve a better regret bound.

Answer of exercise 10.15

1. TRUE: using Beta distributions, we have that the more the parameters α and β are large, the more the distribution is peaked around the empirical mean $\frac{\alpha}{\alpha+\beta}$.
2. TRUE: the update is based on the fact that we have only successes and failures. A version of the TS for different types of rewards can be designed but requires some modifications w.r.t. to its traditional formulation.
3. FALSE: it relies on the sampling on posterior distributions, therefore it resorts to randomization at each round.

Answer of exercise 10.16

1. FALSE: the UCB1 is based on the Hoeffding inequality which holds for limited-support rewards.

2. FALSE: it also might be that we have high uncertainty about its value but turn out to have small expected reward.

3. TRUE: having the same reward distribution in each state is equivalent to say that we have a single state, assuming that all the actions can be performed in each state. Another option is to run a MAB algorithm where each arm corresponds to specific policy to converge to the optimal one over a finite set.

4. FALSE: in generale we need a pair of conjugate distributions to make it work. (I am also accepting as an answer that during the course we only analysed the algorithm with bernoulli distributions)

5. FALSE: this is prevented from the theorem on the lower bound for the regret of stochastic MAB setting, providing a lower bound of log(T).

6. TRUE: the upper bound has a summation over the arms, therefore if we increase the number of arms to consider, we also increase linearly the regret.

7. TRUE: the lower bound holds on average and on generic MAB problems. For specific realizations or for specific MAB settings we might get a smaller regret than the one prescribed by the bound.

8. TRUE: for the stochastic setting. If we have an adversarial setting we require to have enough randomization in the arm selection. (I am also accepting a FALSE if the comment is that we do not have any theoretic result on the regret for such algorithms)

9. TRUE: it relies on the selection of a sample from a posterior distribution, therefore is intrinsically randomized.

Answer of exercise 10.17

1. FALSE/TRUE: if we use the Beta prior, the conjugate distribution is the Bernoulli, thus the rewards should be 0,1. Instead, if we use a Gaussian prior, its conjugate is Gaussian itself and it is possible to execute the TS algorithm with Gaussian rewards.

2. TRUE: it relies on the selection of the largest arm upper bound, whose computation is deterministic.

3. FALSE: the best option to exploit the prior information is the use of Bayesian algorithm, that for the MAB setting is the use of TS.

4. TRUE: it has been shown that their upper bound on the regret is dependent on log(T), where T is the time horizon.

5. FALSE: this is prevented from the theorem on the lower bound for the regret of stochastic MAB setting, providing a lower bound of log(T).

Answer of exercise 10.18

The total number of rounds is: $T = 12 + 24 + 13 + 35 + 16 = 100$

1. The pseudo regret is $T\mu^* - \sum \mu_{i_t}$, where μ^* is the optimal expected reward and μ_{i_t} is the expected reward of the arm chosen for turn t. Similarly, one might compute it as: $T\mu^* - \sum_i (alpha_t + beta_t - 2) * \mu_i$

   The pseudo-regret is: $60.0 - (7.2 + 7.2 + 5.2 + 17.5 + 3.2) = 60.0 - 40.3 = 19.7$

2. The upper bounds computed with UCB1 are:

$$
\begin{aligned}
a_0 : U_t &= 7/12 + \sqrt{(2 \cdot \log(100)/12)} = 1.459 \\
a_1 : U_t &= 7/24 + \sqrt{(2 \cdot \log(100)/24)} = 0.911 \\
a_2 : U_t &= 5/13 + \sqrt{(2 \cdot \log(100)/13)} = 1.226 \\
a_3 : U_t &= 17/35 + \sqrt{(2 \cdot \log(100)/35)} = 0.999 \\
a_4 : U_t &= 3/16 + \sqrt{(2 \cdot \log(100)/16)} = 0.946
\end{aligned}
$$

   Hence, the algorithm is going to choose arm 0 with U = 1.459

3. An adversary knows which kind of algorithm the agent is applying, hence, it can run the same algorithm in parallel to understand what the agent is willing to do next. It will always manage to obtain a loss equal to 1 with UCB1, since it is deterministic. However, with TS, the adversary cannot know in advance which samples would be extracted, but, it can still estimate the probability of each arm to be chosen, knowing the distributions. Therefore, it can choose to put a 1 on the least probable action, in order to maximize the expected instantaneous regret the agent will incur.

Answer of exercise 10.19

1.
   - The cumulated reward is: $0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 1 = 2$
   - The regret is: $9 - 2 = 7$
   - The pseudo-regret is: $0.5 + 0 + 0.2 + 0.2 + 0.5 + 0.2 + 0 + 0.2 + 0.5 + 0 = 2.3$ ($\Delta_1 = 0.5$, $\Delta_2 = 0$, and $\Delta_3 = 0.2$)

2. $U_{1,5} = \frac{0}{2} + \sqrt{\frac{\log(5)}{2}}$, $U_{2,5} = \frac{0}{1} + \sqrt{\frac{\log(5)}{1}}$, and $U_{1,5} = \frac{0}{2} + \sqrt{\frac{\log(5)}{2}}$, therefore, the

arm chosen for the next round would be $a_t = 2$. Consequently, the algorithm run above cannot be the UCB1.

3. Since the TS has a stochastic nature, there is the chance that any possible sequence of the arms is chosen, even if with a small probability. Therefore, the algorithm run above might be TS.

Answer of exercise 10.20

1. The code is implementing an Upper Confidence Bound based algorithm to solve the stochastic MAB problem. It uses the `criterion` variable (i.e., the upper confidence bound) to select the arm (Line 2), and, after observing the reward, it updates the number of pulls of the specific arm (Line 5), the expected payoff (Line 6) and, finally, it computes the bound (Line 9). This process is repeated over a time horizon of T rounds.

2. The algorithm is not correct since the numerator of the term under the square root should be `2 * np.log(t)`. Instead, this version of the bound would increase that term too much over the execution of the algorithm.

3. A viable option for the stochastic MAB problem is the Thompson Sampling algorithm, which adopts a Bayesian approach to solve the MAB scenario. The version of the algorithm we have been shown during the lectures requires that the reward are extracted from a Bernoulli distribution, otherwise the prior-posterior conjugate trick would not apply. Moreover, the other strong assumption is that the environment is stationary over time, otherwise the stochastic MAB techniques would not apply.

Answer of exercise 10.21

1. The lower bound for a stochastic MAB problem is (for T sufficiently large):

$$L_T \geq \log T \frac{\Delta_i}{\sum_{i,i \neq *} KL(\mu_i, \mu^*)},$$

where μ^* is the expected reward of the optimal arm, μ_i is the expected reward of the i-th arm, and $\Delta_i = \mu^* - \mu_i$. In our case we have:

$$\begin{aligned} L_T &> 10 \left(\frac{0.6 - 0.1}{KL(0.1, 0.6)} + \frac{0.6 - 0.3}{KL(0.3, 0.6)} \right) \\ &= 8 \cdot 10 \left(\frac{0.5}{(0.1 \cdot (-1.8) + 0.9 \cdot 0.8)} + \frac{0.3}{(0.3 \cdot (-0.7) + 0.7 \cdot 0.6)} \right) \approx 23.5 \end{aligned}$$

2. The formula for the upper bound of the UCB1 algorithm is:

$$L_T \leq 8 \log T \sum_{i \neq *} \frac{1}{\Delta_i} + 1 + \frac{\pi^2}{3} \left(\sum_{i \neq *} \Delta_i \right),$$

which in our setting becomes:

$$\begin{aligned} L_t &\leq 10 \left(\frac{1}{0.6 - 0.1} + \frac{1}{0.6 - 0.3} \right) + (1 + \frac{\pi^2}{3})(0.6 - 0.1 + 0.6 - 0.3) \\ &= 80(2 + \frac{10}{3}) + (1 + \frac{\pi^2}{3}) * 0.8 \approx 430. \end{aligned}$$

3. All the above theoretical results are only holding in expected value. Both bounds can be violated on specific runs of the UCB1 algorithm.