Multi-Agent Systems

Homework Assignment 4

MSc AI, VU

E.J. Pauwels

Version: November 25, 2024— Deadline: Wednesday, December 4, 2024 (23h59)

4 Fictitious Play

Consider the following pay-off matrix for a 2-player simultaneous game (Capitals indicate the actions, small letters the probabilities with which the corresponding action is played in a mixed strategy):

	$W(w)$	$X(x)$	$Y(y)$	$Z(z)$
$A(a)$	$1,5$	$2,2$	$3,4$	$3,1$
$B(b)$	$3,0$	$4,1$	$2,5$	$4,2$
$C(c)$	$1,3$	$2,6$	$5,2$	$2,3$

Since this game has no pure Nash equilibrium (check this), it must have at least one mixed Nash equilibrium. Recall that $a + b + c = 1$ and $w + x + y + z = 1$.

1. Program the fictitious play algorithm to find a mixed Nash equilibrium.
2. Do the results make sense to you, i.e. can you – *post hoc, i.e. knowing which probabilities seem to be non-zero* – theoretically explain the experimental result? Provide a brief discussion.

5 Monte Carlo simulation

5.1 Recap

Recall that Monte Carlo sampling allows us to estimate the expectation of a random function by sampling from the corresponding probability distribution. More precisely, if $f(x)$ is a 1-dim (continuous) probability density, and $X \sim f$ is a stochastic variable distributed according to this density f, then the expected value of some function φ can be estimated using Monte Carlo sampling by:

$$E_f(\varphi(X)) \equiv \int \varphi(x)f(x)\,dx \approx \frac{1}{n}\sum_{i=1}^{n}\varphi(X_i) \qquad \text{for sample of independent } X_1, X_2, \ldots, X_n \sim f.$$

Simulated p-value In the same vein, if you've observed a specific value for φ_{obs} and you need to decide whether this value is *exceptional* (in some sense) rather than typical, you can compute the *simulated p-value* which quantifies how exceptional that observed value φ_{obs} is in the simulated sample $\varphi(X_1), \varphi(X_2), \ldots, \varphi(X_n)$.

5.2 Warming up ...

1. Assume that $X \sim N(0,1)$ is standard normal. Estimate the mean value $E(\cos^2(X))$. Quantify the uncertainty on your result.

5.3 Quantifying the significance of an observed correlation

2. Suppose you're designing a deep neural network that needs to maximize some score function S. The actual design of the network depends on some hyperparameter A. Training the networks is computationally very demanding and time consuming, and as a consequence you have only been able to perform ten experiments to date. Based on these ten data points you observe a slight positive correlation of 0.3 between the value of the hyperparameter A and the score S. If this result is genuine, it suggests to increase A in the next experiment in order to improve the score. But if the correlation is not significant, increasing A could lead you astray. How would you use MC to decide whether the correlation is significant?
   Hint: Compute the simulated p-value of the observed result, under the assumption of independence.

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

Thompson sampling is an interesting alternative for UCB but requires some introduction. In what follows we will focus on **binary rewards**, so every draw, or pull of the arm, yields either a reward of 1 ($r = 1$ which happens with probability p), or a reward of 0 ($r = 0$ with probability $1-p$). Initially, we know nothing about the success probability p but after some experimentation (pulling the arm and observing the rewards) our uncertainty over p decreases. Beta-distributions are a convenient way to model this change in uncertainty. Loosely speaking, beta-distributions can be seen as a versatile **class of probability densities for (success-)probabilities**. In the next sections we will make this more precise.

6.1 Preliminaries: Beta distributions modeling uncertainties about probabilities

The K-armed bandit problem with binary rewards Consider a bandit that for each pull of an arm, produces a binary reward: $r = 1$ (with probability p) or $r = 0$ (with probability $1-p$). Assuming the bandit has K arms, this means that there are K unknown probabilities $p_1, p_2, \ldots, p_K$ and we need to identify the arm that has the highest probability

$$p^* = \max\{p_1, p_2, \ldots, p_K\},$$

as pulling this arm will result in the highest cumulative reward.

Modeling the success probability of a single arm Let's focus on a single arm, for which the success probability is denoted as p. Initially we have no information (total uncertainty) about the value of p, but each pull of the arm yields a binary outcome (reward), providing some information about p, and thus reducing the corresponding uncertainty. In terms of the probability distribution for p this means that we start with a uniform distribution over the interval $[0, 1]$, but over time the density start peaking over the actual value for p.

Using beta-distributions to model the uncertainty on a probability The **beta-distribution** (cf. `https://en.wikipedia.org/wiki/Beta_distribution`) provides a convenient and mathematically tractable model that captures the behaviour explained above. Specifically, the beta-distribution is a (unimodal) probability distribution on the interval $[0, 1]$ which depends on two parameters: $\alpha, \beta \geq 1$. The explicit distribution is given by (for α, β integers!):

$$B(x;\, \alpha, \beta) = \frac{(\alpha+\beta-1)!}{(\alpha-1)!\,(\beta-1)!} x^{\alpha-1}(1-x)^{\beta-1} \qquad \text{(for } 0 \leq x \leq 1\text{)}.$$

The parameters α and β determine the shape of the distribution. In fact, it is helpful to think of $\alpha - 1$ and $\beta - 1$ as the number of observed successes ($\alpha - 1$) and failures ($\beta - 1$), respectively.

- If $\alpha = \beta = 1$ then we have the uniform distribution. Indeed, lacking any observations, all possible values for p are equally likely.

- If $\alpha = \beta$ the distribution is symmetric about $x = 1/2$. Again, if we have observed an equal number of successes and failures, then $p = 1/2$ is most likely.

- If $\alpha > \beta$ the density is right-leaning (i.e. concentrated in the neighbourhood of 1). This makes sense as observing more successes than failures makes higher values of p more likely. In fact, one can compute the mean explicitly:

  $$X \sim B(x;\, \alpha, \beta) \quad \Longrightarrow \quad EX = \frac{\alpha}{\alpha+\beta} = \frac{1}{1+(\beta/\alpha)},$$

  indicating that the ratio of failures to successes (β/α) determines the position of the mean.

- Larger values of α and β produce a more peaked distribution. Again, this ties in with the intuition that more trials reduce the uncertainty over the outcome, resulting in a more peaked density. Unsurprisingly, this also follows from the formula for the variance:

  $$X \sim B(x;\, \alpha, \beta) \quad \Longrightarrow \quad Var(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

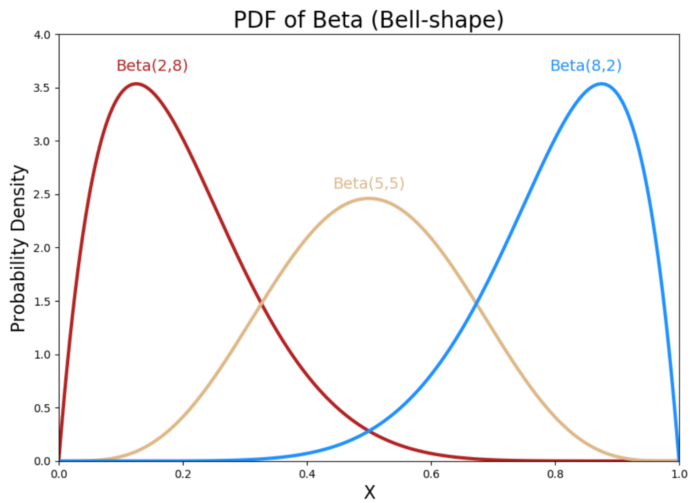

Figure 1: Some probability densities for the Beta-distribution with different parameters.

6.2 Thompson's Bayesian update rule

Although the beta-distribution seems like a reasonable model to quantify the uncertainty on a probability, there is a deeper reason for its use. Updating a (prior) beta-density with binary observations, results in a new beta-density with updated parameters (this is an example of what is known as **conjugated priors**). Specifically, if the **prior** is modeled as $B(x, \alpha, \beta)$, and we observe s successes (1) and f failures (0) then the **posterior** would be the beta distribution $B(x; \alpha{+}s, \beta{+}f)$. This observation yields the rationale for Thompson's Bayesian update rule:

- Initialise $\alpha = \beta = 1$ (yielding a uniform distribution, reflecting our lack of knowledge regarding the value of p). Now repeat the following loop:
  1. Sample from the bandit and get reward r (either $r = 1$ or $r = 0$);
  2. Update the values for α and β as follows:
     - if $r = 1$, then $\alpha \leftarrow \alpha + 1$
     - if $r = 0$, then $\beta \leftarrow \beta + 1$

     This update rule can be summarized as:

$$\alpha \leftarrow \alpha + r \qquad\qquad \beta \leftarrow \beta + (1 - r)$$

Questions

1. Implement the Thompson update rule for single arm bandit (i.e. $k = 1$) and show experimentally that the Beta-density increasingly peaks at the correct value for p. To this end, plot both the evolution of the mean and variance over (iteration)time.

6.3 Thompson sampling for K-armed bandit Problem

For binary outcomes, the Thompson update rule offers an alternative for the UCB-based balancing of exploration and exploitation. Specifically, suppose we have a K-armed bandit problem. The

k-th arm delivers a reward $r = 1$ with (unknown!) probability p_k (and hence $r = 0$ with probability $1 - p_k$). For each arm ($k = 1, \ldots, K$), the uncertainty about the corresponding p_k is modelled using a Beta-distribution $B(x; \alpha_k, \beta_k)$. Thompson sampling now tries to identify the arm that will deliver the maximal cumulative reward (highest p_k) by proceeding as follows:

Initialise all parameters to 1: $\alpha_k = 1 = \beta_k$; Now repeat the following loop:

- **Simulate** We use the beta-distributions to simulate the pulling of each arm. This means that we sample a value U_k from each of the K Beta-distributions:

$$U_k \sim B(x; \alpha_k, \beta_k) \qquad (k = 1 \ldots K).$$

- **Select** Determine which arm gave the best simulated result:

$$k_{max} = \arg\max\{U_1, U_2, \ldots, U_K\}$$

  Mindful of the uncertainties on the p_k-values, the above simulation gives us reason to believe that pulling the k_{max} arm is optimal (after all, we did a simulation using the available evidence, and this was the result).

- **Act** Sample the corresponding arm (i.e. arm k_{max}) and get reward r (either 1 or 0);

- **Update** Use the Bayesian update rule for the corresponding parameters:

$$\alpha_{k_{max}} \leftarrow \alpha_{k_{max}} + r \qquad \text{and} \qquad \beta_{k_{max}} \leftarrow \beta_{k_{max}} + (1 - r)$$

Questions

2. Write code to implement Thompson sampling for the above scenario when $K = 3$;

3. Perform numerical experiments in which you compare Thompson sampling with the UCB. Use **total regret** (provide the precise definition that you're using) as your performance criteria. For UCB, experiment with different values of the hyperparameter c. The fact that, for Thompson sampling, you don't need to specify an hyperparameter, is a distinct advantage.