Problem Set #2: Quantitative Models of Behavior

Hsieh Yu-Guan

March 21, 2017

Introduction

In this report we will work on several classic behavior models. The topics that we will cover include conditional learning (classical conditioning and operant conditioning), decision making and reinforcement learning.

1 Rescola-Wagner model

1.1 Model description

Let's start with the Rescola-Wagner model. It's a model of classical conditioning in which learning signifies building associations between conditioned (CS) and unconditioned (UCS) stimuli. UCS are often represented in form of rewards like food (for an animal) or money (for a person) while CS are some kinds of neutral stimuli that may or may not allow us to predict the occurrence of this reward.

In a single trial, the presence or absence of the reward is respectively denoted by $r = 0$ or $r = 1$. More than one CS can be taken into account, then the presence of the i^{th} stimulus is denoted by $u^{(i)} = 1$ and its absence by $u^{(i)} = 0$. The animal's prediction v is given by the formula

$$v = \sum_{i=1}^{m} w^{(i)} u^{(i)}$$

where m is the number of CS, and for each i, $w^{(i)}$ is the prediction parameter associated with $u^{(i)}$. If we note $u = (u_1, ..., u_m)$ and $w = (w_1, ..., w_m)$, it can also be written in the form

$$v = w \cdot u$$

where $\cdot$ denotes the scalar product. We can now calculate the predition error $\delta = r - v$ which, with the learning rate ϵ, allows us to write down the update rule for every $w^{(i)}$ after the trial at time t

$$w_{t+1}^{(i)} = w_t^{(i)} + \epsilon \delta_t u_t^{(i)}.$$

1.2 A simple test

To examine the reliability of the model, we'll first look at a very simple experiment. Only one stimulus is considered, so we identify for the moment u with $u^{(1)}$ and w with $w^{(1)}$. Let's assume that in the first 25 trials, both stimulus and reward are present, and during the next 25 trials, only the stimulus is present. The plot of r and u is shown in the figure below.

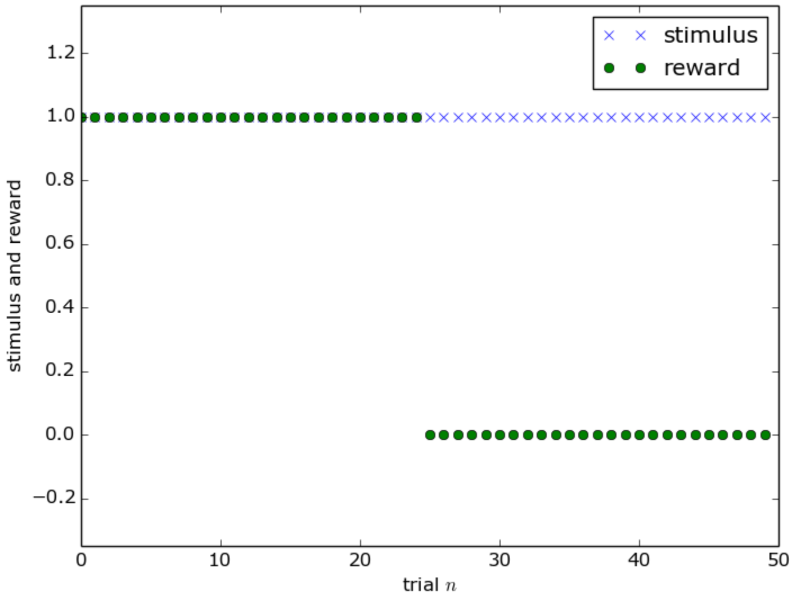

Figure 1: A very simple experiment

Now if we do the simulation with the learning rate $\epsilon = 0.1$, the value of w evolves as shown (since u equals always 1 here, we have also all the time $v = w$).

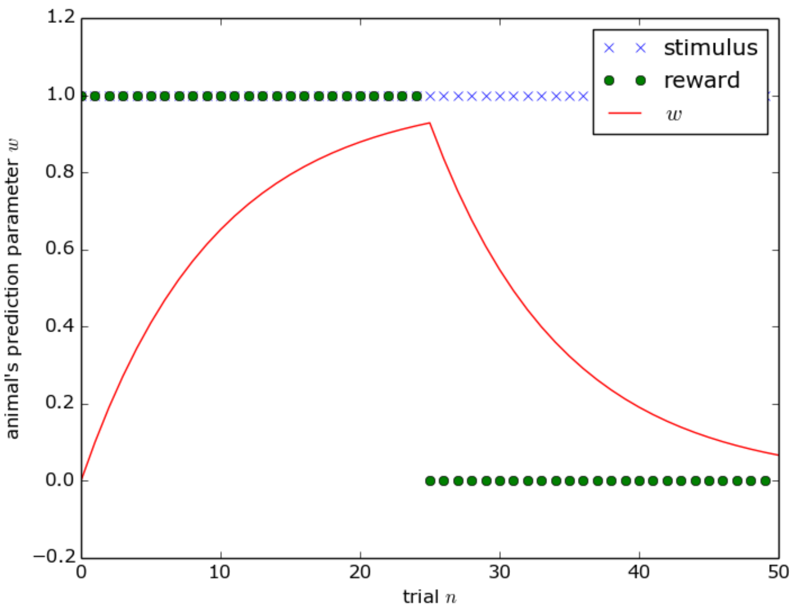

Figure 2: Evolution of w in this simple experiment ($\epsilon = 0.1$)

We see an exponential rise of w to value 1 during the first 25 trials and its value decays exponentially to 0 for the rest of the experiment. It's rather reasonable intuitively, and from a mathematical point of view, we write simply $w_{t+1} - w_t = \epsilon\delta_t u_t$. We know that $\delta_t = r_t - w_t$ (remember that $v_t = w_t$ here) and $u_t = 1$ for all t. For the first 25 trials, we get $w_{t+1} - w_t = \epsilon(1 - w_t)$, so if we put it in a continuous form, it becomes

$$\frac{dw}{dt} = \epsilon(1 - w).$$

We now just need to solve this differential equation to see that $w_t = 1 - e^{-\epsilon t}$. In the same way, for the trials 26 to 50, we can get $w_t = Ae^{-\epsilon(t-25)}$ where A is a constant that can be decided given a particular value of w_t.

The next thing to do is surely to study the impact of the learning rate, so we vary the value of ϵ.

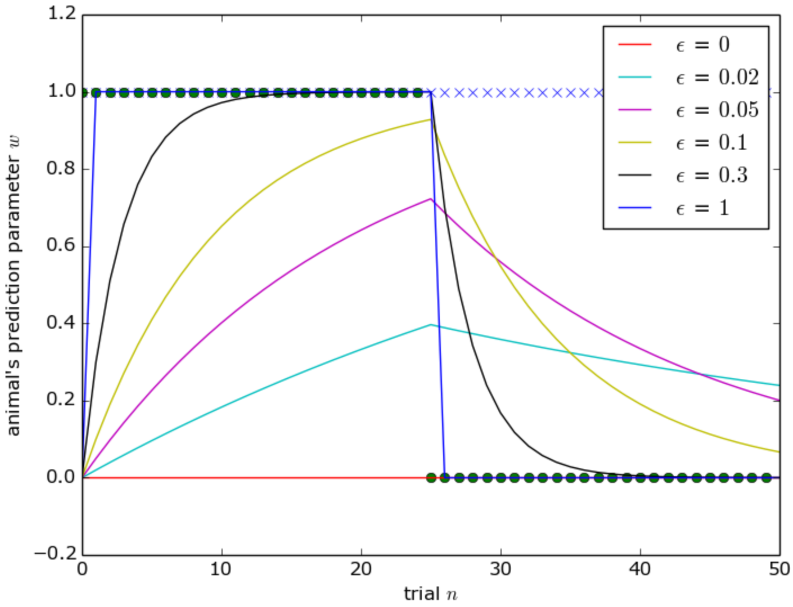

FIGURE 3: Evolution of w in the same experiment but with different ϵ

As what our formula implies, greater the learning rate, faster the animal learns and unlearns the association. It's also interesting to notice that in most of the cases, it's more difficult to unlearn than to learn because the initial δ is smaller (in terms of the absolute value). However, even though one learns faster with a greater learning rate, it doesn't necessarily means that it's better. In fact, there are always noises in what is observed and a smaller learning rate indicates that one is somehow doing an average with past experiences.

Two extreme cases are shown in the figure, when $\epsilon = 0$, the animal can never learn, and when $\epsilon = 1$, the animal exploit only the information from the current trial to compute w and is hence very sensitive to noises. In cosequence, the best value of ϵ depends on the environment where one is.

1.3 Partial conditioning

In this paragraph we are interested in the partial conditioning experiment. We modify slightly the experiment of the last paragraph, there is always only one stimulus and it's all the time present, but the presence of the reward is now a random event with a fixed probability p. We simulate this experiment with $p = 0.4$ and the learning rate $\epsilon = 0.1$ over 160 trials.

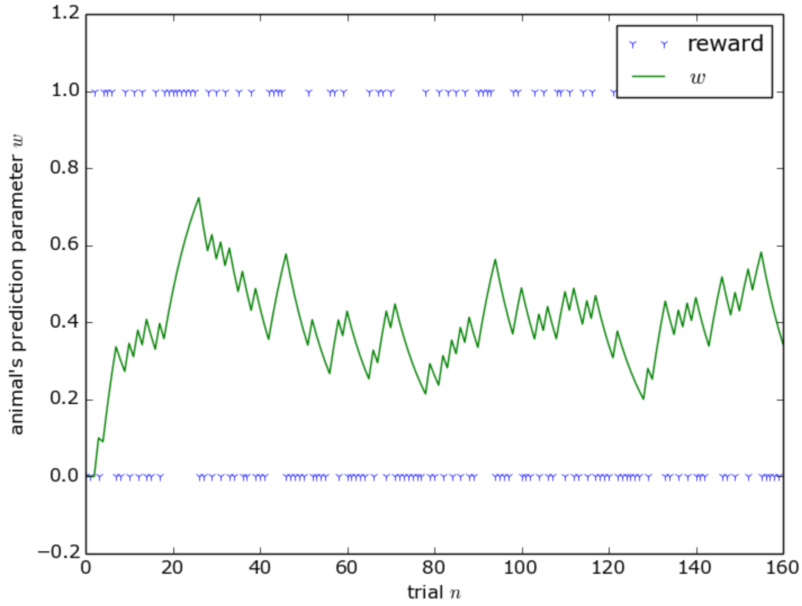

Figure 4: Evolution of w in the partial conditioning experiment ($p = 0.4$, $\epsilon = 0.1$)

We can see that the curve becomes quite noisy because the experiment is not deterministic anymore, but roughly speaking, w tends to increase at the beginning and then oscillates around 0.4. However, its value will never converge. As a matter of fact, the learning rate $\epsilon = 0.1$ is too high and a small number of trials can affect the animal's prediction parameter. We can redo the same simulation but with other ϵ values and over more trials to be able to see the evolution of w for smaller learning rate.

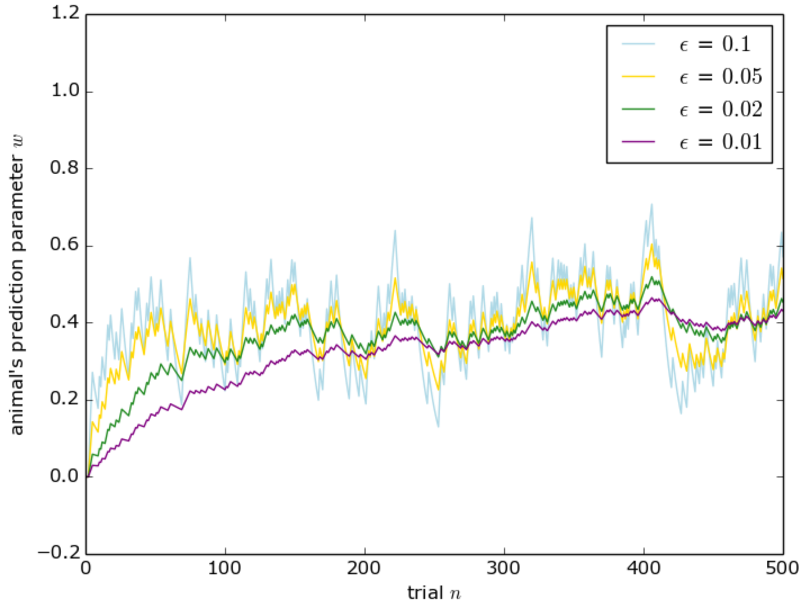

Figure 5: Evolution of w in the partial conditioning experiment with different ϵ ($p = 0.4$)

As predicted, when the learning rate decreases, in particular when it equals 0.01, it takes more time for the animal to learn, but the final value is also more stable and doesn't deviate very much from 0.4. On the contrary, we can imagine that if ϵ is bigger than 0.1, the curve becomes even noisier and one can never learn the probability value $p = 0.4$, which confirms what is said before.

1.4 Blocking effect

Another advantage of the Rescola-Wagner model is that it allows us to explain the blocking effect. It means that the conditioning of an association between a CS and an US can be impaired if during the conditioning process, the CS is presented together with another CS that is already associated with the US. Therefore, in the newer experiment, we need two stimuli CS1 and CS2. In the first 25 trials, only CS1 and the reward (US) are present, and during the next 25 trials, CS1, CS2 and the reward are all present. We choose as usual $\epsilon = 0.1$.

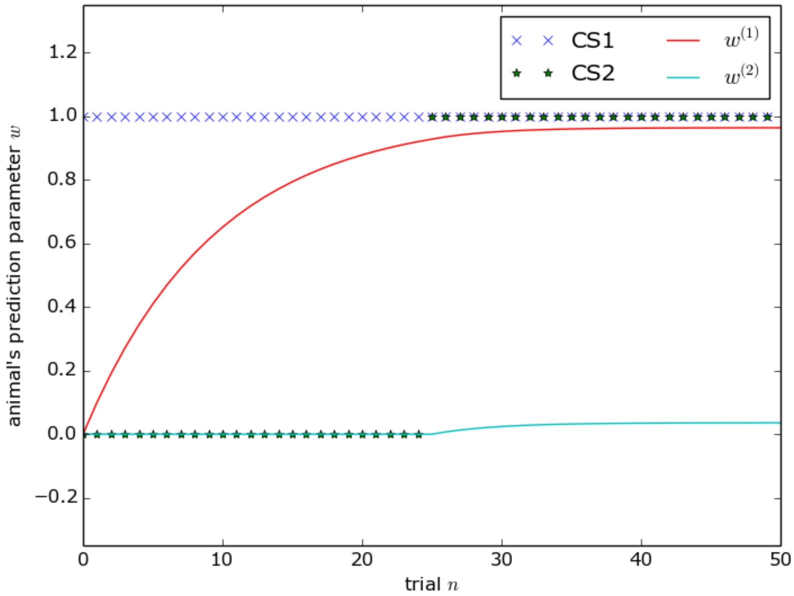

FIGURE 6: Simulation of the blocking effect ($\epsilon = 0.1$)

We get the expected result: the conditioning of CS2 is blocking by the presence of CS1. In the R-W model, the blocking effect can be explained by the fact that $\delta = r - v = r - w^{(1)} - w^{(2)}$ is small during the last 25 trials and using the update rule $w_{t+1}^{(2)} = w_t^{(2)} + \epsilon\delta_t u_t^{(2)}$ we can hardly change the value of $w^{(2)}$.

1.5 Overshadowing

In reality, there is no reason to assume that ϵ is the same for all the stimuli. We should replace the global ϵ by some individual $\epsilon^{(i)}$ for each i. Then, in order to compare different $\epsilon^{(i)}$, a simple experiment can be considered: all the stimuli and the reward are all the time present and we just need to see which particular stimulus is the most associated with the reward after a certain number of trials. For example, if we use two stimuli CS1 and CS2 with respective learning rate 0.2 and 0.1, we plot the result as below.

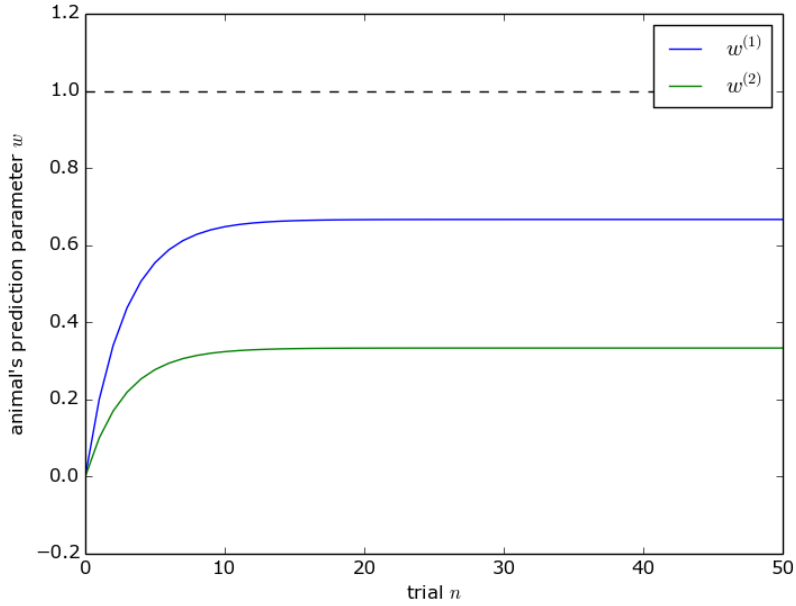

FIGURE 7: Simulation of the overshadowing effect ($\epsilon^{(1)} = 0.2$, $\epsilon^{(2)} = 0.1$)

The stimulus with a higher learning rate (in this case CS1) is more associated with the reward. And if the difference between two learning rates become even larger, it can result in a very weak conditioning of CS2. This is known as the overshadowing effect. We note that neither of the stimuli is fully associated with the reward, but if we add $w^{(1)}$ and $w^{(2)}$, we get something very close to 1.

1.6 Conclusion

In the Rescola-Wagner model conditioning means learning association between conditioned and unconditioned stimuli. At the beginning, we saw that this model could help us understand how one gets conditioned by a stimulus and how this conditioning can again disappear. The most appropriate value of the learning rate may vary from case to case. Next, it turned out that even in a non-deterministic experiment the model is able to find the key probability value. Blocking and overshadowing can also be explained.

Of course this simple model has its limit. For instance, high-order conditioning requires us to take into consideration the time factor, which is not done yet for the time being. Nonetheless, this model is not totally absurd either. In fact, studies have suggested that the activity of some dopamine neurons in the brain encodes effectively the prediction error δ of the model.

2 Operant conditioning

2.1 Model description

Operant conditioning is also called instrumental conditioning. It differs from classical conditioning in that the acquired reward or punishment is mainly decided by the agent's behavior, so what one needs to learn is the association between each behavior and its consequence.

We'll illustrate this idea through a small example here. A bee is collecting nectar from yellow and blue flowers, and at every moment, each type of flower carries a specific reward, which is denoted by r_b for blue flowers and r_y for yellow ones. However, the bee is not aware of the exact values of nectar rewards. Instead, it has some internal estimates m_b and m_y. What the bee needs to do is therefore to make decisions on the basis of these two values and to do real-time updates of them using what it knows.

2.2 Softmax strategy

For the decision part, we assume that the bee adopts the softmax strategy. That is, it chooses the flower of type i with probabily

$$p_i = \frac{\exp(\beta m_i)}{\exp(\beta m_y) + \exp(\beta m_b)}$$

where $i \in \{b, y\}$ and β is the "expoitation-exploration trade-off" parameter (or the inverse temperature parameter if we refer to the Boltzmann distriution model). Writing in this form, it can be easily generalized to situations with more than two types of flowers, but we can also write, for example for p_b

$$p_b = \frac{1}{1 + \exp(\beta(m_y - m_b))}.$$

The name of β comes from the fact that it controls the bee's attitude towards exploration and exploitaton behavior. To see this, we can fix $m_y - m_b$ and plot p_b as a funcion of β.

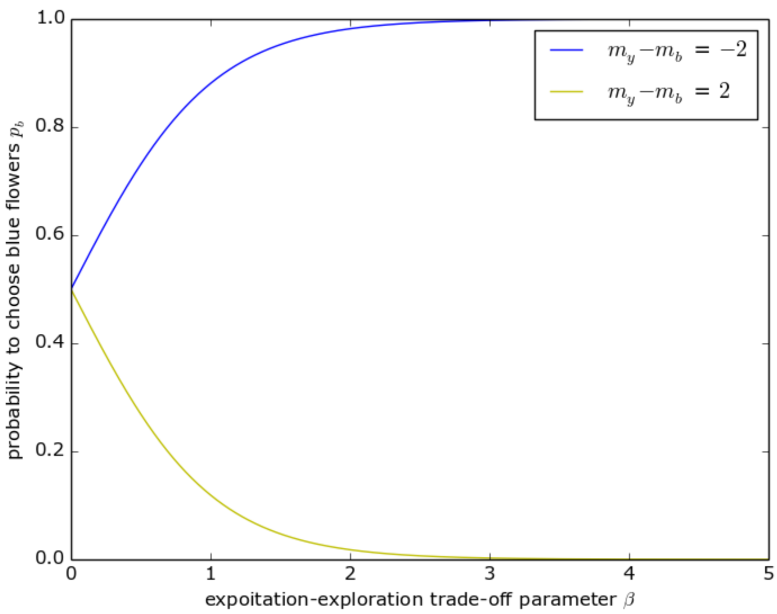

FIGURE 8: Plot of p_b as a funcion of β with fixed $m_y - m_b$

When the difference $m_y - m_b$ is positive, yellow flowers are more rewarding to the bee than blue flowers, so generally speaking it tends to go to yellow flowers to collect nectar. This tendancy is however not very clear when β is small, it suggests that the bee doesn't trust very much its own estimates and puts emphasis on the exploration side.

On the other hand, when β gets larger (in this case typically when it's greater than 2), the bee goes to yellow flowers almost all the time. This implies the bee exploits a lot what it has learned and doesn't explore much. Now if $m_y - m_b$ is negative, it means that blue flowers are more attractive to bees so the two curves are horizontally symmetrical but the effect of β is the same. We can also plot p_b as a function of $m_y - m_b$ by fixing β.

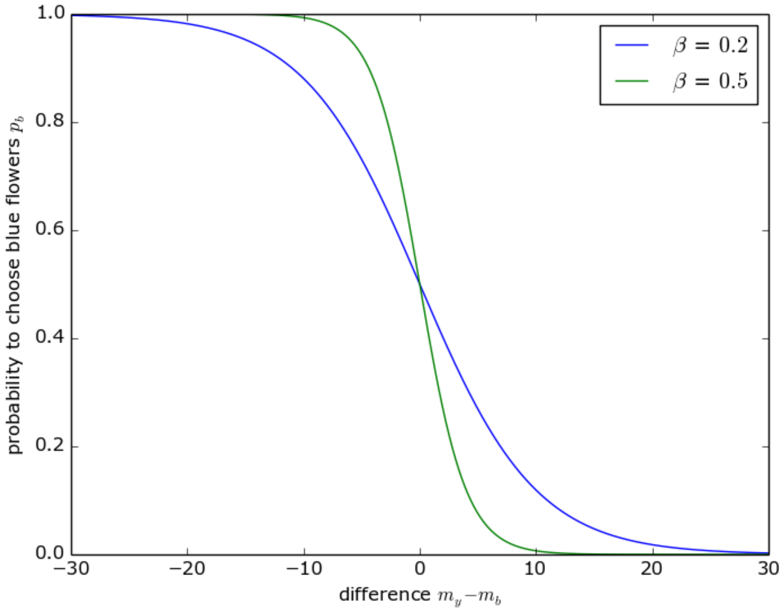

FIGURE 9: Plot of p_b as a funcion of $m_y - m_b$ with fixed β

It's not surprising to get some sigmoid curves whose steepness depends on the parameter β. When $m_y - m_b$ tends to minus infinity, p_b tends to 1 because the bee believes that blue flowers are much better than their yellow counterparts, and when $m_y - m_b$ tends to plus infinity, p_b tends to 0 (here we're supposing implicitly $\beta > 0$).

2.3 Dumb bee

Before discussing how to update estimated rewards, we'll first do some simulations with the policy given in the last paragraph. The simulation is for two days. During the fisrt day we have $r_b = 8$ and $r_y = 2$. During the second day, they're set to $r_b = 2$ and $r_y = 8$. The bee is able to sample 100 flowers during one day. Throughout the simulation, the bee can never learn and believes that $m_y = 5$ and $m_b = 0$. First let's assume $\beta = 0$.

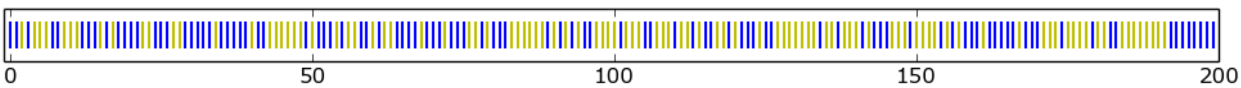

FIGURE 10: The choice of the bee for 200 samplings in two day, $\beta = 0$

It's known that when $\beta = 0$ the bee chooses blue and yellow flowers equiprobably regarless of its internal estimaates. This is indeed what is observed here, it's not obvious to say the bee goes to blue or flowers more often (naturally, blue bars for blue flowers and yellow bars for yellow flowers). We do one more simulation with this time $\beta = 0.8$.

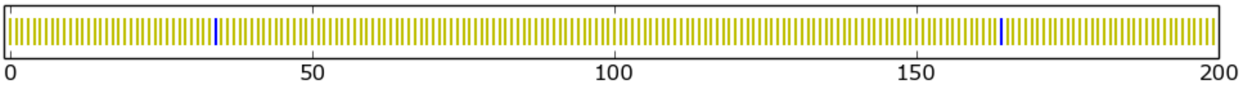

FIGURE 11: The choice of the bee for 200 samplings in two day, $\beta = 0.8$

In this case using the given formula we have $p_b = 1/(1+e^4) \sim 1.7\%$. Our result of the simulation is not very far from this value, the bee visits blues flowers only twice which stands for a probability of 1%. Anyway, in the two cases, we can say that the bee's performance is quite poor because it doesn't learn from experiences, in average the reward that the bee gets is 5 but it could have done better.

2.4 Smart bee

To learn the estimated reward, we inspire from the first part of the report, the R-W model. Consequently, the online update rules are given by

$$m_b \to m_b + \epsilon(r_b - m_b)$$
$$m_y \to m_y + \epsilon(r_y - m_y)$$

where as usual ϵ is the learning rate. The first rule is only used when the bee visits a blue flower and similarly the second rule is considered only when it visits a yellow flower. We start again from $m_y = 5$, $m_b = 0$ and we give $\epsilon = 0.2$.