# Lecture 10
# Conditioning and Reinforcement

jochen.braun@nat.uni-magdeburg.de

May 25, 2005

**Credits:**

Dayan & Abbott, Chapter 9

Schultz (2000) Nat Rev Neurosci

Schultz, Dayan, Montague (1997) Science

# 1 Classical conditioning

Following Dayan & Abbott, we model how animals learn to expect a reward and frame the discussion in terms of (conditioned) stimuli, rewards, and expectation of reward.

## 1.1 Rescorla-Wagner rule

The Rescorla-Wagner rule provides an economical account for many aspects of classical conditioning. Each stimulus is associated with a weight that predicts the value of the associated reward. The weights are learned by comparing the predicted/expected reward with that actually received.

The presence or absence of multiple stimuli over time is represented by a binary-valued vector $\mathbf{u}(t_i)$ (assuming discrete times $t_i$). The expected reward is computed as a linear sum with weights $\mathbf{w}(t_i)$:

$$v(t_i) = \mathbf{w}(t_i) \cdot \mathbf{u}(t_i)$$

The weights are adjusted trial-by-trial by the Rescorla-Wagner rule

$$\mathbf{w}(t_{i+1}) = \mathbf{w}(t_i) + \epsilon\,\delta\,\mathbf{u}(t_i) \qquad\qquad \delta = r(t_i) - v(t_i)$$

The crucial term is the prediction error $\delta$. $\epsilon$ is the learning rate, which is a measure of the associability of stimulus and reward. The RW rule changes $\mathbf{w}$ systematically until the average value of $\delta$ is zero. At this point $\mathbf{w}$ fluctuates about the equilibrium value $\mathbf{w} = \langle \mathbf{u}\, r \rangle$

There are two types of instrumental conditioning tasks. In the first type, the reward is delivered immediately after the action is taken, making learning relatively easy. In the second type, the reward depends on an entire sequence of actions and is thus delayed until the sequence is completed. This makes learning much more difficult.

The Rescorla-Wagner rule and linear reward prediction summarize a wide range of animal learning data for classical and instrumental conditioning with immediate rewards. The limitations of this approach become apparent in situations such as 'secondary conditioning' where rewards are delayed.

Secondary conditioning involves associating a first stimulus with a reward and then a second stimulus with the first stimulus (without any reward). This causes the second stimulus to become associated with a reward (although it has never been paired with it). The RW rule cannot account for this observation.

## 1.2 Temporal difference learning

To formulate a scheme that predicts future (and possibly delayed) rewards, we can substitute time-dependent functions for what previously were fixed values. In other words, we allow the stimulus $\mathbf{u}(t)$, the reward $r(t)$ , the learned weights $\mathbf{w}(t)$, the expected reward $v(t)$ to change over the course of each trial. We assume that trial time $t$ is discrete and takes values from 0 to $T$. Our goal is an algorithm that makes $v(t)$ equal to the total future reward expected, at time $t$, until the end of the trial.

$$v(t) \approx \left\langle \sum_{\tau=0}^{T-t} r(t+\tau) \right\rangle$$

Generalising the linear prediction used before to time-dependent weights $\mathbf{w}(\tau)$ (which predict a reward at time $t$ on the basis of a stimulus at time $t-\tau$) gives us

$$v(t) = \sum_{\tau=0}^{t} \mathbf{w}(\tau) \cdot \mathbf{u}(t-\tau)$$

$$\mathbf{w}(\tau) \rightarrow \mathbf{w}(\tau) + \epsilon\, \delta(t)\, \mathbf{u}(t-\tau) \qquad\qquad \delta(t) = \sum_{\tau=0}^{T-t} r(t+\tau) \quad - \quad v(t)$$

In practise, we cannot compute $\delta(t)$ in this way because we do not yet know future rewards $r(t+1)$, $r(t+2)$, and so on. However, assuming that $v(t)$ really does approximate the total future reward, we can estimate $\delta(t)$ from the *temporal difference* in $v(t)$ over successive timesteps:

$$\delta(t) \;=\; \sum_{\tau=0}^{T-t} r(t+\tau) - v(t) \;=\; r(t) + v(t+1) - v(t)$$

This approach is called temporal-difference learning.

As the peak in $\delta$ moves backwards from the time of the reward to the time of the stimulus, weights $w(\tau)$ progressively grow. This gradually extends the prediction of future reward $v(t)$ from an initial transient at the time of the stimulus to a broad plateau extending from the time of the stimulus to the time of the reward. Eventually, $v$ predicts the time of reward delivery by dropping to zero.

Temporal difference learning can account for *secondary conditioning* (when a first stimulus becomes associated with a reward and a second stimulus with the first stimulus, without any reward). Suppose an association between stimulus $s_1$ and a future reward has been established. When a second stimulus $s_2$ is introduced before the first stimulus, the positive spike in $\delta(t)$ at the time of $s_1$ gradually increases the weight associated with stimulus $s_2$ and thus establishes a positive association between stimulus $s_2$ and the reward.

## 1.3 Dopamine and reward prediction

The neural substrate for the prediction error $\delta$ may be dopaminergic neurons in the ventral tegmental area (VTA) and the substantia nigra. Schultz and colleagures have recorded dopamine neurons in monkeys and found phasic activation to unexpected rewards (fruit juice). After repeated pairing of predictive stimuli (light) with delayed reward( fruit juice), the phasic activation shifted to the time of the predictive stimulus. If the expected reward did not occur, a phasic suppression was observed. This behaviour was observed in the majority of VTA dopamine neurons.

## 2 Instrumental conditioning and static action choice

In natural settings, rewards are perhaps more often associated with actions taken than with stimuli observed. Animals are very adept at developing strategies that increase reward. We first consider *static action choice*, where the reward immediately follows the action taken. The particular example examined by Dayan & Abbott is **bee foraging**.

To formalise the situation, we assume that the bee chooses blue or yellow flowers with a certain probability $P[b]$ and $P[y]$, respectively, and as a result receives a reward $r_b$ or $r_y$ drawn from probability densities $p[r_b]$ or $p[r_y]$. To allow for a probabilistic (rather than deterministic) choice on the part of the bee, we assume that

$$P[b] = \frac{1}{1 + \exp\left[\beta(m_y - m_b)\right]} \qquad\qquad P[y] = \frac{1}{1 + \exp\left[\beta(m_b - m_y)\right]}$$

where $m_b$ and $m_y$ are *action values* that determine the frequency with which blue and yellow flowers are visited and that must be adjusted by learning and $\beta$ is a constant that determines the variability of the bee's actions. For small $\beta$, the bee spends more time *exploring* and, for large $\beta$, the bee behaves almost deterministically and spends its time *exploiting.*

We consider two methods of solving the bee foraging task, the *indirect actor* where the bee learns to predict nectar volumes of specific flower times and the *direct actor* where the bee learns to predict average nectar harvest.

### 2.1 Indirect actor

Choosing on the basis of expected nectar volumes means setting the action values to $m_b = \langle r_b \rangle$ and $m_y = \langle r_y \rangle$. This can be achieved by the Rescorla-Wagner, or delta, rule:

$$\begin{array}{rcc} \text{Choose blue} & m_b \to m_b + \epsilon\,\delta & \delta = r_b - m_b \\ \text{Choose yellow} & m_y \to m_y + \epsilon\,\delta & \delta = r_y - m_y \end{array}$$

## 2.2 Direct actor

An alternative approach is to maximize the total expected reward, rather than the expected rewards for visiting each type of flower. In this case the action values are logically distinct from the average reward values. This will turn out to be useful later on, when we adjust action values and expected rewards with two concurrent but separate learning procedures. The learning goal can be formulated as follows:

$$\langle r \rangle = P[b]\, \langle r_b \rangle + P[y]\, \langle r_y \rangle$$

$$\frac{\delta \langle r \rangle}{\delta m_b} = \beta\, \left(P[b]\, P[y]\, \langle r_b \rangle - P[y]\, P[b]\, \langle r_y \rangle\right) \qquad \frac{\delta \langle r \rangle}{\delta m_y} = \beta\, \left(P[y]\, P[b]\, \langle r_y \rangle - P[b]\, P[y]\, \langle r_b \rangle\right)$$

$$\frac{\delta \langle r \rangle}{\delta m_b} = \beta\, P[b]\, (1 - P[b])\, (\langle r_b \rangle - \bar{r}) \;-\; \beta\, P[y]\, P[b]\, (\langle r_y \rangle - \bar{r})$$

$$\frac{\delta \langle r \rangle}{\delta m_y} = \beta\, P[y]\, (1 - P[y])\, (\langle r_y \rangle - \bar{r}) \;-\; \beta\, P[b]\, P[y]\, (\langle r_b \rangle - \bar{r})$$

The maximization can be achieved by *stochastic gradient ascent*, that is, by choosing changes in parameters $m_b$, $m_y$ such that, on average, they are proportional to $\delta \langle r \rangle / \delta m_b$ and $\delta \langle r \rangle / \delta m_y$, respectively. The expressions above are formulated with such a procedure in mind. $\bar{r}$ is an arbitrary that controls the variability of learning steps and thus the speed of learning. Reformulated as a gradient ascent prescription this gives:

$$\begin{array}{rl} m_b \rightarrow m_b + \epsilon\, (1 - P[b])\, (r_b - \bar{r}) & \quad \text{if b selected} \\ m_b \rightarrow m_b - \epsilon\, P[b]\, (r_y - \bar{r}) & \quad \text{if y selected} \\ m_y \rightarrow m_y + \epsilon\, (1 - P[y])\, (r_y - \bar{r}) & \quad \text{if y selected} \\ m_y \rightarrow m_y - \epsilon\, P[y]\, (r_b - \bar{r}) & \quad \text{if b selected} \end{array}$$

Note that we have used:

$$P[b] = \tfrac{1}{1+\exp[\beta(m_y - m_b)]} \qquad \tfrac{\delta P[b]}{\delta m_b} = \beta\, P[b]\, P[y] \qquad P[y] = \tfrac{1}{1+\exp[\beta(m_b - m_y)]} \qquad \tfrac{\delta P[y]}{\delta m_b} = -\beta\, P[b]\, P[y]$$

## 3 Sequential action choice

If choice behaviour is not based on immediate information (i.e., immediate consequences of actions), the situation is much more difficult. This is illustrated well by the *maze task*, where a hungry rat moves through a maze starting from point A. When it reaches a termination point, the rat is rewarded by certain number of food pellets and is removed from the maze. Then the rat starts again at A. Its objective is to maximize the total amount of food received. In this case, it can achieve this objective by moving left at A and right at B.

The difficulty is that neither choice at A is rewarded immediately. If the rat goes left at A and at B, it has to learn that the first choice was good but the last choice bad. In other words, the reward for going left at A is delayed until the rat goes right at B.

A well known method for optimizing control problems such as the maze task is *policy iteration*, where a stochastic policy (which determines action choices at each decision point) is maintained and iteratively improved. One element is a *critic* that uses temporal difference learning to estimate the total future reward expected from the current policy. Another element is an *actor* which maintains and improves the policy. In this approach the rat learns the appropriate action at A using the same methods that it uses to learn it at B or C (where the reward is immediate). This it can do by using the expected future reward as a reinforcement signal.

## 3.1 Policy evaluation by "critic"

In this step, the rat keeps its policy fixed and uses temporal difference learning to learn the total future reward associated with each location.

The simplest possibility is a random choice of left and right. In this case, inspection of the maze shows the expected total future reward to be

$$v(B) = \frac{0+5}{2} = 2.5 \qquad v(C) = \frac{0+2}{2} = 1 \qquad v(A) = \frac{v(B)+v(C)}{2} = 1.75$$

These values can be learnt using the temporal difference learning rule. If the rat chooses action $a$ at location $u$ and ends up at location $u'$, we modify the weights $w(u)$ as follows:

$$v(u) = w(u) \qquad w(u) \to w(u) + \epsilon\,\delta \qquad \delta = r_a(u) + v(u') - v(u)$$

where $r_a(u)$ is the immediate reward (if any) of taking action $a$ at location $u$, $v(u)$ is the expected total future reward, $w(u)$ is the learnt weight, and $\epsilon$ is a parameter that controls learning rate.

## 3.2 Policy improvement by "actor"

To improve policy, *action values* $m_a(u)$ for taking action $a$ at location $u$ are adjusted. This happens on the basis of the expected total future rewards rather than immediate rewards. Strictly speaking, policy evaluation should therefore already be completed.

The expected worth of action $a$ at location $u$ is the immediate reward $r_a(u)$ (if any) plus the expected worth $v(u')$ at the next location $u'$. The average worth across all actions that can be taken at $u$ is simply $v(u)$. The difference between these two is the $\delta$ that drives learning of both the critic (previous section) and the critic (here):

$$\delta = r_a(u) + v(u') - v(u)$$

$$\begin{aligned} m_l(u) &\rightarrow m_l(u) + \epsilon\,(1 - P[l,u])\,\delta && \text{if l selected} \\ m_l(u) &\rightarrow m_l(u) - \epsilon\,P[l,u]\,\delta && \text{if r selected} \\ m_r(u) &\rightarrow m_r(u) + \epsilon\,(1 - P[r,u])\,\delta && \text{if r selected} \\ m_r(u) &\rightarrow m_r(u) - \epsilon\,P[r,u]\,\delta && \text{if l selected} \end{aligned}$$

$$P[l,u] = \frac{1}{1 + \exp\left[\beta(m_r(u) - m_l(u))\right]} \qquad P[r,u] = \frac{1}{1 + \exp\left[\beta(m_l(u) - m_r(u))\right]}$$

For example at location A, $\delta$ takes the following two values (if policy evaluation is perfect):

$$\begin{aligned} \delta = 0 + v(B) - v(A) &= 0.75 && \text{for left turn} \\ \delta = 0 + v(C) - v(A) &= -0.75 && \text{for right turn} \end{aligned}$$

The learning rule makes actions with $\delta > 0$ more likely and actions with $\delta < 0$ less likely. In this case, it would increase the chance that the rat makes the (correct) left turn. Note that the learning rule improves policy at every step, leading to a monotonic increase in the expected future reward.

If we interleave policy evaluation and policy improvement steps, the approach miraculously converges to produce a happy result.

## 4 Application to water maze

In the water maze task, rats are placed in milky water and have to find a small submerged platform. After several trials, the rats learn the location and swim directly to it.

A possible model starts with the activity of 'hippocampal place cells' that encode the current position in the pool. The model also assumes eight possible actions at every moment (corresponding to compass directions) and a representation of the expected total future reward.