CHAPTER

10

Bayesian Approaches to Learning and Decision-Making

Quentin J.M. Huys[1,2]

[1] University Hospital of Psychiatry, Zürich, Switzerland; [2] University of Zürich and Swiss Federal Institute of Technology (ETH), Zürich, Switzerland

OUTLINE

Computational Psychiatry
http://dx.doi.org/10.1016/B978-0-12-809825-7.00010-9

10.1 INTRODUCTION

Learning and decision-making are highly intertwined processes. While learning influences what decisions are taken, the decisions taken determine what will be learned. Jointly, they serve the purpose of optimizing behavior and a breakdown in either will upset the functioning of the other. This vicious circle is often seen in mental illness, where poor decisions in mental illness lead to the self-selection of individuals into high-risk situations (Kendler et al., 1999) and thereby likely to more mental illness.

In this chapter, we will consider a series of approaches to the guidance of behavior. Some, mostly from Reinforcement Learning (RL; Sutton and Barto, 1998) involve "learning," while others, from the related field of Dynamic Programming, are more akin to inference (Bertsekas and Tsitsiklis, 1996). The key aspect to consider is that actions taken now do not just have rewarding or punishing consequences now, but also in the future. For instance, theft may lead to a short-term gain, but in the longer term may well lead to very significant losses that far outweigh the short-term gains. Identifying optimal behaviors at any one point in time, therefore, requires thinking ahead and considering the various possible consequences of any current behavior. This, however, is extremely difficult: first, the list of possible things that may happen in the future is vast, and second, the future is uncertain. RL is a field with a host of techniques for taking long-term outcomes into account when making decisions.

This chapter will first introduce so-called Markov Decision Problems (MDPs) and their solutions formally. In a second part, it will give the reader tools to use these models to examine choice behavior. In a third part, we will examine a few specific models as examples of decision-making in health and illness. In the following, we focus on the key concepts and omit a number of important details for the sake of simplicity. The interested reader is referred to Bertsekas and Tsitsiklis (1996) and Sutton and Barto (1998) for accessible but more in-depth treatments.

10.2 MARKOV DECISION PROBLEMS

Fig. 10.1A shows the general MDP setup that underlies RL and Dynamic Programming methods. An MDP is defined by five components that we will briefly introduce below:

- set of states $s \in \mathcal{S}$
- set of actions $a \in \mathcal{A}$ and an associated set of action transition matrices $\mathcal{T}^a$
- reward function $\mathcal{R}$
- policy π

The intuition is that an agent is in some particular state s. In this state, the agent can perform certain actions a. Depending on the environment, this leads to a new state s' and a reinforcement r, which can be positive or

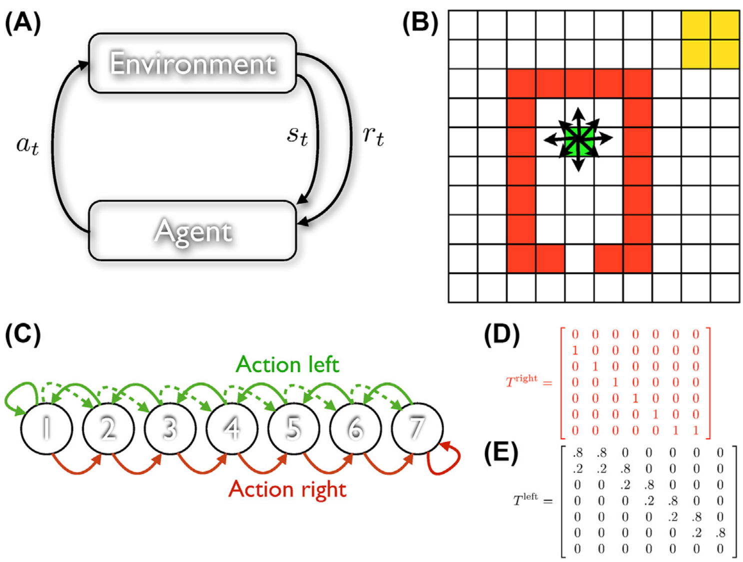

FIGURE 10.1 (A) The setting. An agent interacts with an environment by choosing actions which result in rewards and in turn influence its current state. (B) Grid world example. Each square in the grid is a different state s. The state of the agent is indicated by a *green square*, i.e., it is roughly in the middle of the grid. Actions correspond to moving around on this grid. In this example, the agent can move to all adjacent squares, i.e., has 8 actions available in each state (exemplified by the *black arrows* emerging from the *green square* in the middle). Some states lead to losses, here indicated by the color red, and some to gains, here indicated by yellow. A policy assigns each state preferences for particular actions. The aim is to find an optimal policy, i.e., one that maximizes long-term rather than just immediate reward. (C) Simple linear state space with two actions. While the red action "right" is deterministic and thus has only 0s and 1s in the transition matrix (D), the green action left is probabilistic, corresponding to a transition matrix with off-diagonal terms (E).

negative. Fig. 10.1B shows a more specific example: a so-called grid world, where the state is simply the position on the grid.

The techniques described below will typically focus on simple definitions of states within particular experiments, where the relevant states can simply be the stimuli presented during the experiment. However, the notion of state s in RL is potentially very broad. In neuroscience terms, it could include internal states such as arousal or hunger, and as such is clearly a very complex construct.

The actions a are defined in terms of their impact on states. In Fig. 10.1C, the action "going left" is defined in terms of moving from any one state to its left neighbor. More generally, actions are defined in terms of probability distributions over successor states (Fig. 10.1D and E). Putting all state succession probabilities for one action, next to each other, into one matrix results in the transition matrix $\mathcal{T}^a$ for that action (Fig. 10.1D and E). This describes the consequences of emitting that action in each of the existing states; it is generally assumed that the transition matrices are fixed and determined by the world, though they may not be known to the agent.

This definition of actions has an important consequence for how states are defined: The consequences of actions must depend only on the current state, and not on past states. Consider braking when driving a car. The impact of braking depends not only on the position of the care, but also on its speed. Hence, the impact of braking on transitions to other states cannot be described purely in terms of the current position. For the techniques below to apply, the problem must be a so-called MDP. For this to be true, speed should be part of how states are defined in the car example, such that the consequence of braking is clearly defined for each state independent of what the previous states were.

The reward r is a scalar, i.e., a unidimensional number that takes on positive or negative values for rewards and losses, respectively. The richness of real rewards is captured by the dependence on actions and state transitions: Rewards r are generated by a reward function $\mathcal{R}(s, a, s')$ that depends both on the action taken, and the current and next states. Just like ingesting food is rewarding when hungry but not when sated, taking a step to the right can lead to a loss in states left of the red punishing barrier in Fig. 10.1, and to reward when left of the yellow reward area. Just like the transition matrices $\mathcal{T}$, the reward function $\mathcal{R}$ is assumed to be a fixed part of the environment, though again it may not be known to the agent. The agent's estimates of the transition matrices and the reward function are referred to as the agent's *model* $\mathcal{M}$ of the world.

The aim is to find an optimal policy $\pi^*(a; s)$. A policy $\pi(a; s)$ describes the probability of taking an action a in state s. A policy is optimal if it always chooses one of the optimal actions in each state, where the optimal action is the one that maximizes the total sum of rewards that can be earned in the long term. Conceptually the simplest approach to infer the

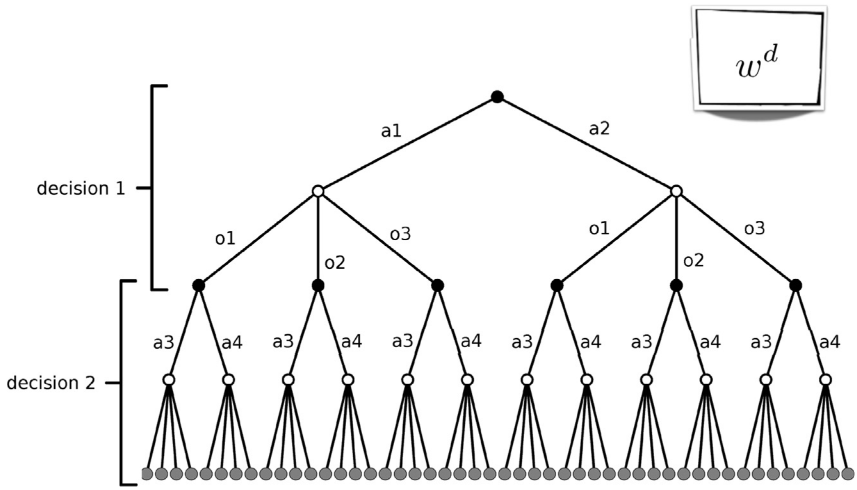

FIGURE 10.2 Decision tree. At the root of the tree, there are two available actions a1 and a2, each of which probabilistically leads to one of three outcomes (o1−o3). For each of these, there are new options a3 and a4. Overall, the size of the tree increases rapidly with the depth d and width w of the tree as w^d.

optimal policy is to consider all possible actions from a state; all the resulting state transitions and rewards; then all possible next actions for the successor states, etc. This results in a decision tree, with the root at the current state (Fig. 10.2). Unfortunately, these decision trees grow rapidly in size. For the simple grid world example, the number of actions and successor state to each state is 9 (disregarding the boundaries), and hence the decision tree corresponding to looking d steps ahead has 9^d branches. Such an explicit tree search is hence prohibitive for all but the very simplest of problems.

10.2.1 Bellman Equation

Optimal, in RL, is defined in terms of achieving the maximal expected sum over rewards $r_{t'}$ in the future, i.e., for times $t' \geq t$. The expected total future reward from state s at time t when following a particular policy π is called the value $\mathcal{V}^\pi(s)$ of the state and defined as:

$$\mathcal{V}^\pi(s_t) = \mathbb{E}\left[\sum_{t'=0}^{\infty} r_{t+t'}\gamma^{t'} \middle| s_t; \pi\right] \tag{10.1}$$

where the discounting factor $0 \leq \gamma < 1$ is not only necessary to ensure that the sum is finite, but also gives rewards in the near future more weight than rewards in the distant future. It is set to 1 if only finite problems are considered. The key insight is that Eq. (10.1) is a sum and due to the linearity of expectations (because the average of two means is the same as the mean of two averages), it can be rewritten into two terms as:

$$\mathcal{V}^{\pi}(s_t) = \underbrace{\mathbb{E}[r_t|s_t;\pi]}_{\text{immediate reward}} + \underbrace{\mathbb{E}\left[\sum_{t'=1}^{\infty} r_{t+t'}\gamma^{t'}\middle|s_t;\pi\right]}_{\gamma\cdot\text{reward from next timestep onwards}}$$

The total future reward from the next timestep onward, the second term in the equation above, is simply the value of the next state–action pair $\mathcal{V}^{\pi}(s_{t+1})$, and hence we can write as:

$$\mathcal{V}^{\pi}(s_t) = \mathbb{E}[r_t|s_t;\pi] + \mathbb{E}\left[\gamma\mathcal{V}^{\pi}(s_{t+1})\middle|s_t;\pi\right]$$

The rewards r_t are drawn from the reward process $\mathcal{R}(s_t, a_t, s_{t+1})$. The expectations $\mathbb{E}[\cdot]$ are over two processes: first, the likely actions taken, and second, the likely consequences of those actions. Expanding these expectations and substituting the policy π for the first, and the transition matrices $\mathcal{T}$ for the second, results in the so-called Bellman equation (Bellman, 1957; Sutton and Barto, 1998):

$$\mathcal{V}^{\pi}(s_t) = \sum_{a_t}\pi(a_t;s_t)\sum_{s_{t+1}}p(s_{t+1}|a_t,s_t)(\mathcal{R}(s_t,a_t,s_{t+1}) + \gamma\mathcal{V}^{\pi}(s_{t+1})) \quad (10.2)$$

or, using a more compact notation:

$$\mathcal{V}^{\pi}(s) = \sum_{a}\pi_s(a)\sum_{s'}\mathcal{T}^{a}_{ss'}\left(\mathcal{R}^{a}_{ss'} + \gamma\mathcal{V}^{\pi}(s')\right)$$

10.2.2 Solving the Bellman Equation

Eq. (10.2) describes a consistency between values of states s and its successor states s' for a given policy π. If the reward function $\mathcal{R}$ and transition matrices $\mathcal{T}$ are known, then this consistency can be used to solve the equation and infer the values $\mathcal{V}^{\pi}(s)$ for all states s. The first and conceptually most straightforward way is to recognize that Eq. (10.2) is linear and can be rewritten in vector form. Dropping the subscript t and letting the successor state be s', we have:

$$[\mathbf{v}^{\pi}]_s = \mathcal{V}^{\pi}(s)$$

$$[\mathbf{r}^{\pi}]_s = \sum_{a}\pi(a;s)\sum_{s'}p\left(s'\middle|a,s\right)\mathcal{R}(s,a,s')$$

$$[\mathbf{T}^{\pi}]_s = \sum_{a}\pi(a;s)\sum_{s'}p\left(s'\middle|a,s\right)$$

We can now rewrite the Bellman equation as

$$\mathbf{v}^\pi = \mathbf{r}^\pi + \gamma \mathbf{T}^\pi \mathbf{v}^\pi \tag{10.3}$$

which is simply solved by:

$$\mathbf{v}^\pi = (\mathbf{I} - \gamma \mathbf{T}^\pi)^{-1} \mathbf{r}^\pi$$

Here, we note an important feature of the effective transition matrix $\mathbf{T}^\pi$ induced by the policy. It is a square stochastic matrix all columns of which are probability distributions. As such, its leading eigenvector is 1, and the steady-state distribution of state visits is the eigenvector corresponding to that leading eigenvalue. The values are hence only finite as long as $\gamma < 1$. An alternative is to have a matrix $\mathbf{T}^\pi$, the leading eigenvector of which <1. This is true if all states have a finite probability of leading to an absorbing state that cannot be left and which has zero reward. This latter setting effectively curtails the infinite sum of rewards in Eq. (10.1) to a finite sum of exponentially distributed length.

A different approach to solving the Bellman equation is to note that if the values assigned to states are incorrect, then there is a difference Δ between the left and the right side of Eq. (10.3):

$$\Delta = \mathbf{r}^\pi + \gamma \mathbf{T}^\pi \mathbf{v} - \mathbf{v}$$

This can be used to turn the Bellman equation into an update equation:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + \Delta_i \\ &= \mathbf{r}^\pi + \gamma \mathbf{T}^\pi \mathbf{v}_i \end{aligned} \tag{10.4}$$

which can be shown to converge to the true value $\mathbf{v}^\pi$ for the same reason as above (Bertsekas and Tsitsiklis, 1996).

10.2.2.1 Model-Free Temporal Difference Prediction Error Learning

These previous approaches to evaluating the value function require the model $\mathscr{M}$ of the world consisting of the transition matrices $\mathcal{T}$ and the reward function $\mathcal{R}$ to be known, and are hence instances of "model-based" value estimation. So-called model-free techniques do not require this. Instead, they only require that samples can be drawn from the transition matrix and the reward function. Drawing samples corresponds to observing the reward and state consequences of taking an action, i.e., drawing an action $a_t \sim \pi(a; s_t)$ given the current state s_t; and then observing a successor state $s_{t+1} \sim p(s_{t+1}|a_t, s_t)$, and a reward $r_t \sim \mathcal{R}(s_t, a_t, s_{t+1})$ (see Fig. 10.1A). The Bellman equation (Eq. 10.2) contains two expectations, one over the transition probabilities and one over the action probabilities, which can be approximated with samples drawn from the two distributions. Temporal difference learning effectively performs the iterative update of Eq. (10.4) after every sample, but includes a learning rate $0 \leq \alpha \leq 1$:

$$\begin{aligned}\mathcal{V}_{t+1}(s_t) &= \mathcal{V}_t(s_t) + \alpha\delta_t \\ &= \mathcal{V}_t(s_t) + \alpha(r_t + \mathcal{V}_t(s_{t+1}) - \mathcal{V}_t(s_t))\end{aligned} \tag{10.5}$$

This fixed learning rate α effectively induces an exponentially decaying average over past samples. If it is chosen to decay with the number of times a particular state has been sampled, this procedure can be shown to converge to the true value function of the policy over time under some conditions (see toy example below).

10.2.2.2 Phasic Dopaminergic Signals

Notably, the long-term expected future reward can be learned over time by comparing the expected reward $\mathcal{V}_t(s_t)$ with the sum of the received reward and the expected reward of the successor state $\mathcal{V}_t(s_{t+1})$. The difference between the two, δ_t, is the temporal difference prediction error thought to be reported by phasic dopaminergic firing (Schultz et al., 1997). We note here that this can be positive for a transition from a state of low-reward expectation to a state of high-reward expectation even if the immediate reward is zero. This is thought to explain the transfer of phasic firing observed during conditioning of a cue to predict reward. Early on in learning, dopaminergic neurons do not respond to the cue, but do respond to the (unexpected) reward. Over time, as the animal learns that the cue predicts the reward, the value $\mathcal{V}$ of the cue increases, and its unexpected presentation elicits a prediction error, and hence firing in the dopaminergic neurons. However, as the reward is predicted, the value $\mathcal{V}$ is equal to the reward r, and hence a prediction error no longer occurs at the time of reward, resulting in no dopaminergic firing.

10.2.3 Policy Updates

Given the value $\mathcal{V}^\pi$ of each state under a given behavioral policy π, the policy can now be improved in a very simple manner by choosing that action, which has the highest expected value in each state, i.e.,

$$\pi^{\text{new}}(a;s) = \begin{cases} 1 & \text{if} \quad a = \text{argmax}_{a'} \mathcal{Q}^\pi\left(a', s\right) \\ 0 & \text{else} \end{cases}$$

where

$$\mathcal{Q}^\pi(a_t, s_t) = \sum_{s_{t+1}} p(s_{t+1}|a_t, s_t)(\mathcal{R}(s_t, a_t, s_{t+1}) + \gamma\mathcal{V}^\pi(s_{t+1}))$$

is the state–action $\mathcal{Q}$ value of taking action a_t in s_t under the old policy π. Again, this can be shown to converge to the optimal policy under some conditions (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998). What is notable here, is that optimal policies are always deterministic—there is no reason ever to choose a suboptimal action.

Though conceptually simple, such policy updates are biologically unreasonable, as they would require completely evaluating the value function for a policy before any behavioral adaptation. Updating the policy before having performed a full evaluation of the value function has the potential of breaking many of the guarantees. In contrast, Q-learning (Watkins and Dayan, 1992) is an "off-policy" method. This means that the estimated values are not affected by the sampling process (the policy). It proceeds as follows:

$$\mathcal{Q}_{t+1}(a_t, s_t) = \mathcal{Q}_t(a_t, s_t) + \alpha\left(r_t + \gamma \max_a \mathcal{Q}_t(a, s_{t+1}) - \mathcal{Q}_t(a_t, s_t)\right)$$

The key differences are the maximum operation over the next actions to be taken, which requires some foresight and can be computationally challenging if the potential behavioral repertoire is large. As long as all state–action pairs continue to be sampled, this converges to the true state–action value for any policy, and hence the policy can be updated and learning occur online.

10.3 MODELING DATA

10.3.1 General Considerations

Having provided a brief overview over the key features of RL and dynamic programming, we now turn to a tutorial overview of how these techniques can be used to probe human (and animal) decision-making. The framework suggested here is distinct from the standard approach in a number of ways. First, it is a generative framework. This means that the model can be run on the experiment under scrutiny and used to simulate data akin to that obtained in the experiment. Rather than modeling only specific aspects of the data, such as the averages in different conditions, the approach is to model the process by which the data came about, and the data itself, in their "holistic" entirety. For this, the internal inference processes that give rise to the data have to be captured in sufficient detail. The result is that learning or inference processes can be tested on the data in their entirety. The test statistics are replaced by parameters determining the internal processes. Unlike traditional test statistics, their meaning is made explicit by their function in the model.

Model building The first step is to build a series of models. Each contains an internal process by which different choice options are valued, and a link function which describes how preferences turn into observed decisions. At least two models should be built: a model M0 of ″no interest″ that performs the task, but without involving the process of interest, and a model M1 that does contain the process of interest.

Validation on surrogate data

1. **Data generation**: Run each model on the experiment from which data will be examined. Do the generated data look reasonable?
2. **Surrogate model fitting**: Fit each model to the data generated from it. Are the true parameters readily recovered? Are some parameters not identifiable?
3. **Surrogate model comparison** : Does the model comparison procedure correctly identify the data generated by each model?

Real data analysis

1. **Real model fitting**: Fit each model to the real data.
2. **Real model validation**: Run each model with the fitted parameters on the exact experimental instance presented to that particular subject. Are the key features of the real data captured reasonably?
3. **Real model comparison**: choose the least complex model that best accounts for the data.
4. **Parameter examination** : only at this point should the parameters of the model be examined, and only the parameters of the most parsimonious model should be ascribed meaning.

FIGURE 10.3 Overview over modeling approach.

The freedom to build different models is huge and vastly extends the kinds of processes that can be inferred and tested. However, as each model has to be built separately, there is also ample scope for a variety of mishaps. As a result, the modeling should contain three general steps. In a first step, the model needs to be built; in a second step, this model should be validated with surrogate data; and in a third step, the model is applied to the real data. A general suggested framework is shown in Fig. 10.3 (Daw, 2009).

A few comments are worthwhile. The key first step clearly is the model building. Here, the valuation processes by which choice preferences arise in the models are the hypotheses to be tested. A reasonable approach is to build a series of models starting from a very simple "null" valuation process, and then adding in the various features of interest to examine to what extent they parsimoniously contribute toward explaining the data. The second component is the link function, which needs to be probabilistic to allow noisy experimental data to be fitted. We noted above that optimal policies are always deterministic. Making this assumption when fitting models makes them very brittle as errors due to other, unforeseen and maybe unrecorded events are interpreted as strong evidence. Hence, one role of the link function is to assimilate noise from a variety of sources, and inferring its parameters allows for individual variation in this. Nevertheless, its functional form should be checked, and we will return to this below.

Validation on surrogate data serves a number of purposes. First, it is important to check that the data the model generates are actually comparable to the data obtained in the experiment. Second, by fitting data from the surrogate model, the ability to identify and recover parameters is established. This is an important step before interpreting any parameters.

Third, the ability to reliably distinguish between different models can be established on surrogate data comparable to the one available in the experiment under scrutiny. Indeed, it is prudent to attempt to perform these steps before running the experiment in real as they may suggest changes in experimental parameters, such as the length of the tasks or the number of subjects to run.

Finally, the models need to also be validated on the actual data under scrutiny. One possibility is to compare data generated from the model (with fitted parameters) to the real data. For learning experiments, it is, for instance, often useful to plot learning curves and ask whether the model captures the shape of these curves well. Once the models have been thus validated, it is meaningful to ask which of the models provide the most parsimonious account of the data. This is the domain of model comparison. Note that a model comparison is always relative and does not provide any absolute information and even the best amongst a set of models may still be too poor to provide any meaningful information. The interpretation of parameters in the models should only follow at the end, once one model has been chosen as a good characterization of the data.

10.3.2 A Toy Example

As a first example, we consider very simple learning experiment in Fig. 10.4A. In this experiment, each action a_t on trial t yields an immediate reinforcement r_t, but does not have any influence on future options. Hence, the total summed future reward in this case is simply the average immediate reward offered by each of the stimuli.

The first model assumes that individuals perform temporal difference learning, adapted to this extremely simple scenario. Taking Eq. (10.5) and observing that there is no next state, but only immediate rewards, the temporal difference prediction error learning becomes simple prediction error learning $\mathcal{V}^{TD}_{t+1}(s_t) = \mathcal{V}^{TD}_t(s_t) + \alpha\big(r_t - \mathcal{V}^{TD}_t(s_t)\big)$, as in Rescorla–Wagner learning (Rescorla and Wagner, 1972). The second model assumes that individuals simply perform averages over the reinforcements earned for each of the two stimuli, which is the correct inference to perform given how the outcomes are generated. The expected values $\mathcal{V}^{av}$ are hence,

$$\mathcal{V}^{av}_{t+1}(s) = \frac{1}{t}\sum_{t'=1}^{t} r_{t'} = \frac{1}{t}\left(\sum_{t'=1}^{t-1} r_{t'} + r_t\right) = \frac{t-1}{t}\mathcal{V}^{av}_t(s) + \frac{1}{t}r_t$$

$$= \frac{1}{t}\big((t-1)\mathcal{V}^{av}_t(s) + r_t\big) + \mathcal{V}^{av}_t(s) - \mathcal{V}^{av}_t(s)$$

$$= \mathcal{V}^{av}_t(s) + \frac{1}{t}\big(r_t - \mathcal{V}^{av}_t(s)\big)$$

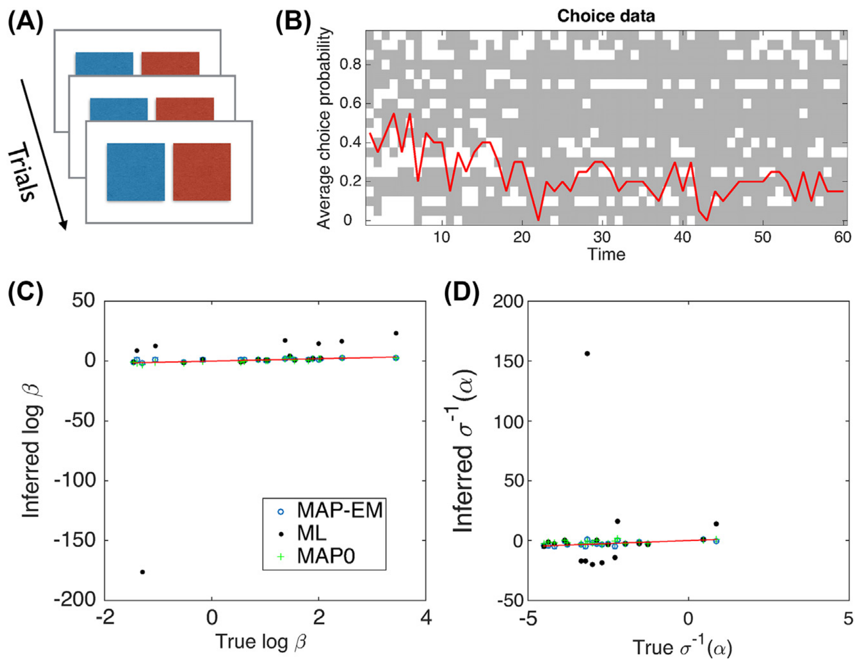

FIGURE 10.4 (A) Simple toy learning experiment. On each trial, individuals have to choose one of two squares. The *blue square* yields small rewards on 80% of trials, and the *red square* on 20% of trials. (B) Surrogate data generated from a simple learning model. Each of the horizontal rows shows the choice data for one subject, with gray indicating choice of the blue and white choice of the *red button*. The *red superimposed line* is the average probability of choosing the *red button* across subjects on that trial. (C) Plots of true parameters β against the parameters inferred from data in panel (B). The *red line* indicates correct equality. (D) Plots of true learning rates α against those inferred from data in panel (B). Note that both parameters were transformed to deal with natural limits on their values: to ensure $\beta \geq 0$ all models are written in terms of $\beta = \exp(\beta')$, and to ensure $0 \leq \alpha \leq 1$ they are written in terms of $\alpha = 1/(1 + \exp(\alpha'))$. *MAP-EM*, maximum a posteriori using expectation maximization to infer the priors; *ML*, maximum likelihood.

The first line rewrites the sum over all past rewards as an iterative update. The second line then rewrites this into a form similar to that of the temporal difference (TD) learning rule. Comparing these, we see that the fixed learning rate α in the TD learning rule has been replaced by a decaying term $1/t$ in the average. While the averaging rule gives each of the t samples the same weight, the TD rule always gives the most recent sample a weight α, and the samples before that an exponentially smaller weight. While the TD rule has one free parameter α, the averaging rule has no free parameters.

10.3.3 Generating Data

Given a model of the choice process, it is straightforward to generate data by using a link function that maps the values $\boldsymbol{\mathcal{V}}$ onto probabilities of taking particular actions. A frequent choice is the use of a softmax link function whereby the probability of choosing stimulus s on trial t is:

$$p(a_t = s|\boldsymbol{\mathcal{V}}_t) = \frac{e^{\beta \boldsymbol{\mathcal{V}}_t(s)}}{e^{\beta \boldsymbol{\mathcal{V}}_t(s)} + e^{\beta \boldsymbol{\mathcal{V}}_t(\bar{s})}} \tag{10.6}$$

The data in Fig. 10.4B were generated from the TD model with this softmax.

10.3.4 Fitting Models

Having built a model and generated data from it, the next step is to fit the model to the generated data. Fitting a model means finding the set of parameters that are most compatible with the data. The maximum likelihood (ML) parameters are those under which the data are most likely. To find them, we must maximize the likelihood of all the T actions $a_1, \dots a_T$ by one subject given that subject's parameters:

$$\widehat{\theta}^{ML} = \underset{\theta}{\operatorname{argmax}} \log p(a_1, \dots a_T|\theta) \tag{10.7}$$

The question is how to compute the total likelihood of all choices. On first sight, this appears difficult because choices depend on previous choices and so cannot be treated independently. However, if every choice only depends on the value $\boldsymbol{\mathcal{V}}_t$ at the time of the choice t, as assumed in Eq. (10.6), then the probability of observing a sequence of stimulus choices $a_1, \dots a_T$ is simply:

$$\log p(a_1, \dots a_T|\theta) = \log \prod_{t=1}^{T} p(a_t|\boldsymbol{\mathcal{V}}_t) = \sum_{t=1}^{T} \log p(a_t|\boldsymbol{\mathcal{V}}_t) \tag{10.8}$$

which is notable: Even though choices at any time t clearly depend on the previous ones; once we condition on the values the choices become independent of the previous choices. The values can be updated iteratively before computing the likelihood of each choice, leading to an algorithm that takes the general and very simple form shown in Algorithm 10.1.

initialize values $\mathcal{V}$
foreach *trial* t **do**
  compute log likelihood of choice a_t on trial t given parameters : $l_t = \log p(a_t|\mathcal{V}_t, \theta)$
  update value $\mathcal{V}_{t+1}$ given outcomes on trial t
end
compute total log likelihood $l = \sum_t l_t$

ALGORITHM 10.1 Likelihood computation.

The total likelihood can now be passed to any of a number of optimization tools to solve Eq. (10.7). Fig. 10.4C and D shows the result of an ML fit in black for the TD model with the two parameters α and β. As can be seen, the black dots are sometimes very far off the diagonal, which unfortunately is relatively typical for these kinds of models. Although ML estimators are asymptotically unbiased, they do have high variance. This is often a prominent problem because parameters often do have overlapping effects and therefore can trade-off each other. In these examples, whenever β was set to a very small value, the learning rate α was set to a very high value.

The blue circles show a very simple and often very powerful solution to this, which is to impose a soft prior on the parameters and performing maximum a posteriori (MAP) inference rather than ML. This is very simply achieved by replacing Eq. (10.7) with

$$\widehat{\theta}^{MAP} = \underset{\theta}{\operatorname{argmax}} \log p(a_1, \ldots a_T | \theta) p(\theta)$$

The computation of the posterior likelihood is thus just the same as that of Algorithm 10.1, but with the log likelihood of the prior added to the total log likelihood of the choices.

At times, however, the choice of the prior $p(\theta)$ can be difficult. In these situations, it can make sense to infer the prior from the data in an empirical Bayesian setting (Huys et al., 2012). There are a number of techniques available for this, and this is becoming a more common approach. Fig. 10.4C and D shows this in blue. For this simple example, little is gained over the basic MAP approach, but this changes for larger models.

10.3.5 Model Comparison

Having fitted the model to the data, we can ask how good an account it provides. When doing so, however, it is not sufficient to simply look at the model fit. Fig. 10.5A shows data generated from a straight line with some noise added. The top panel shows a linear fit, while the bottom panel shows a sixth order polynomial. Clearly the latter is a better fit despite the fact that the top is closer to the truth. To understand why the model with the better fit is nevertheless poorer, consider Fig. 10.5B and C. As the data (orange dots) bunch up toward the right, they are better fit by one of the triangular probability distributions in panel B than by the two uniform distributions in panel C. The model in panel B, is very powerful. Different parameter settings lead to wildly different distributions that often miss the data entirely and predict data which is never observed. Hence, the powerful model is likely to predict novel data less well. We can think of this as a trade-off between the different settings a model allows, and the fit it provides to the data. Fig. 10.5D illustrates that this problem exists for learning models, too.

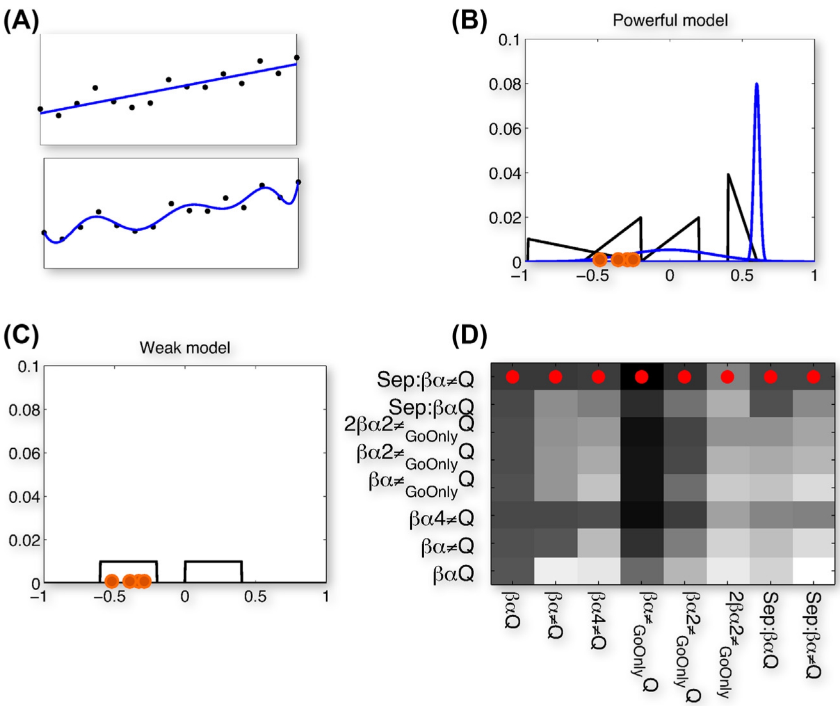

FIGURE 10.5 Model comparison. (A) Data (*black dots*) generated from a straight line with added noise is fit better by a complex sixth order polynomial (bottom) than by a *straight line* (top). This is overfitting. (B and C) Intuition for the need to average over all possible parameter settings to infer a model's parsimony. An overly complex model will contain many parameter settings that provide poor accounts of the data (orange), and only very few that provide a good fit. When averaging these, the many poor fits outweigh the few very good fits (B). Conversely, a simple model may not fit the data so well but is never far from the data and does not predict data that are never observed. (D) Learning data generated from models of increasing complexity (left to right), and fitted with models of increasing complexity. The best fitting model with best likelihood is always the most complex one at the top.

Bayesian model comparison takes this into account by using as a measure of fit not the best possible likelihood, but the average likelihood over all possible parameter settings:

$$p(\boldsymbol{\mathcal{A}}|\mathcal{M}) = \int d\theta p(\boldsymbol{\mathcal{A}}|\theta, \mathcal{M})p(\theta) \tag{10.9}$$

The Bayes factor between two models is then defined as

$$BF = \log_e \frac{p(\boldsymbol{\mathcal{A}}|\mathcal{M}_1)}{p(\boldsymbol{\mathcal{A}}|\mathcal{M}_2)} \tag{10.10}$$

and is considered substantial if greater than 3, and conclusive if greater than 5 (Kass and Raftery, 1995). Unfortunately, the integral in Eq. (10.9) is not always straightforward to evaluate, and there exist a number of approximations to it. The simplest ones are the Akaike Information Criterion $\text{AIC} = -2\log p\left(\mathcal{A}\middle|\hat{\theta}^{ML}\right) + 2d$, and the Bayesian Information Criterion $\text{BIC} = -2\log p\left(\mathcal{A}\middle|\hat{\theta}^{ML}\right) + d\log(n)$, where d is the number of parameters in the model and n is the number of data points. These penalizes models by counting their parameters. AIC tends to be less conservative, while BIC can be too conservative. Another possibility is to perform a Laplace approximation around the MAP parameters (Daw, 2009).

10.3.6 Group Studies

The methods so far have considered individual subjects. However, most studies, particularly in clinical settings, deal with group data. Fig. 10.6 shows different approaches to group data. Two simple approaches are to treat all individuals as using the same parameters, i.e., a fixed-effects treatment (panel A) or treating them entirely separately (B). While the former conflates different types of noise and is therefore not recommended, the latter can inflate noise depending on how the parameters are estimated. A more natural approach is to respect the fact that individuals in a group tend to be similar, and hence should have similar parameters (Fig. 10.6C; Huys et al., 2012). However, even this still assumes that all individuals use the same model. Two relaxations of this approach exist. First, one can employ a random-effects treatment over models (Fig. 10.6D; Stephan et al., 2009), or one can nest multiple models in a more complex model (Fig. 10.6E; Daw et al., 2011; Guitart-Masip et al., 2012). While the former assumes that individuals in a group may differ in terms of their internal processes, it assumes that these internal processes are homogeneous. The latter conversely assumes that individuals employ a mixture of strategies, but that this is true across the entire group. We note that nesting models are problematic in that there can be an overfitting by the more powerful component *within* individuals.

10.4 DISSECTING COMPONENTS OF DECISION-MAKING

Having described the theoretical core of decision-making and how to fit these valuation models to data, we turn to four examples. These are chosen to illustrate some of the insights gained from detailed modeling of behavioral data with a combination of RL and Bayesian techniques.

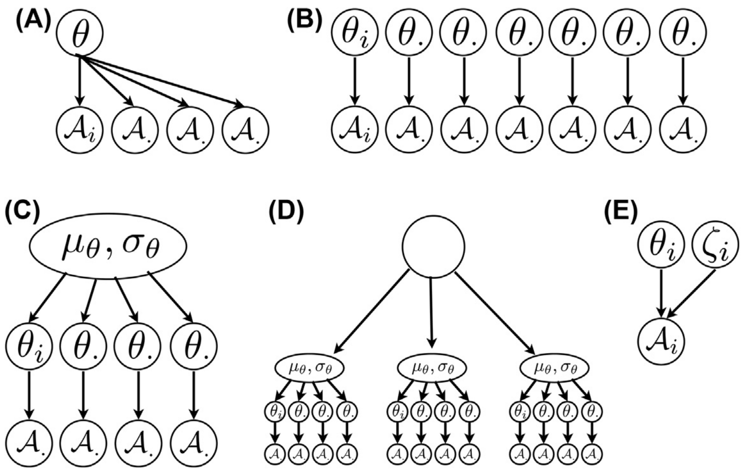

FIGURE 10.6 Group data. (A) A fixed-effects analysis would assume that all subjects share the same parameters. This is not recommended. (B) The extreme opposite is to perform separate maximum likelihood fits for each subject. This in effect assumes that all subjects are independent and have parameters that are not a priori related. (C) In a group design, it is natural to assume that individual subjects are drawn from a group that describes their similarity. For instance, parameters of individuals in a group could cluster around a particular value. However, although this model is a random-effects model in terms of the individual parameters, it is nevertheless still a fixed-effects treatment of the model itself; all individuals are assumed to be examples of the same model. (D) Next, it is possible to consider random-effects treatments of the models, i.e., that some individuals in a group will behave according to model 1, others according to model 2, and yet others according to model 3. (E) Finally, it is possible to examine whether individuals behave according to two different models. As this is simply a more complex model, it can be combined with the approaches in panel (A—D).

10.4.1 Reward Learning

Alterations to how rewards are processed are important in a number of psychiatric conditions. For instance, anhedonia is one of the core elements of depression and refers to an inability to experience pleasure. Pizzagalli et al. (2005) asked whether anhedonia might specifically influence the ability of people with depression to learn from rewards. They used a perceptual decision-making task where subjects had to report the length of a briefly presented mouth (Fig. 10.7A) as either short or long. Unbeknownst to the subjects, one option was rewarded more frequently than the other. Over time, subjects came to express a bias toward identifying the more rewarded stimulus, but this bias was abolished by anhedonia. This task raises two possibilities: either anhedonia blunts the sensitivity to rewards; or it blunts the ability to learn from the rewards. In principle, this might be testable by using a very simple prediction error learning to value the different choices:

$$\mathcal{Q}_{t+1}(a_t, s_t) = \mathcal{Q}_t(a_t, s_t) + \alpha(\rho r_t - \mathcal{Q}_t(a_t, s_t)) \tag{10.11}$$

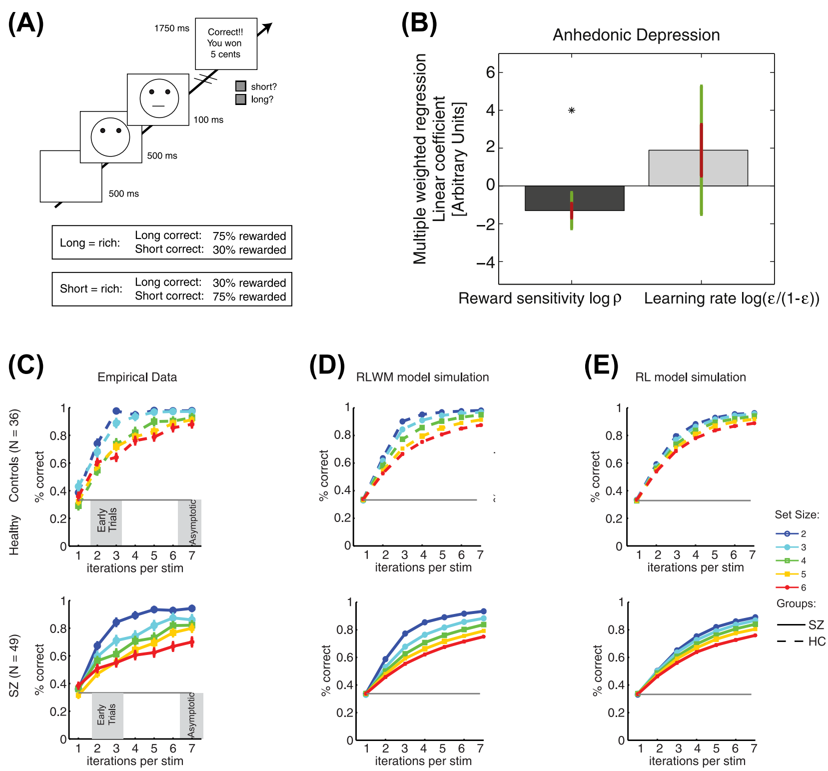

FIGURE 10.7 Reward learning. (A) Pizzagalli et al. (2005) perceptual decision-making task. Subjects have to indicate whether a briefly flashed mouth is long or short. Unbeknownst to them, one option is more frequently rewarded than the other, leading to a bias in reporting that option amongst healthy subjects. However, this bias could arise from either changes in the sensitivity to rewards, or changes in the ability to learn from rewarding events. (B) Across multiple studies using this task, anhedonia was related to reward sensitivity, but not to learning rate. (C) Requiring subjects to learn about multiple stimuli at the same time slows down learning both in controls (top) and patients with schizophrenia (bottom). (D) Including a working memory component in the model accounts for the pattern of data in controls (top); and its impairment for the pattern in patients (bottom). (E) A model without a working memory component is not able to account for the observed patterns. *HC*, hippocampus; *RL*, reinforcement learning; *RLWM*, RL model with working memory; *SZ*, patients with schizophrenia. *Panels (A and B) reproduced from Huys, Q.J.M., Pizzagalli, D.A., Bogdan, R., Dayan, P., 2013. Mapping anhedonia onto reinforcement learning: a behavioural meta-analysis. Biol. Mood Anxiety Disord. 3 (1), 12; Panels (C–E) from Collins, A.G.E., Brown, J.K., Gold, J.M., Waltz, J.A., Frank, M.J., 2014. Working memory contributions to reinforcement learning impairments in schizophrenia. J. Neurosci. 34 (41), 13747–13756.*

where ρ scales the size of the received reward, while α is the learning rate. However, as alluded to above, this can be rewritten as:

$$\mathcal{Q}_{t+1}(a_t, s_t) = (1-\alpha)^t \mathcal{Q}_0(a_t, s_t) + \alpha\rho \sum_{t'=0}^{t} (1-\alpha)^{t'} r_{t-t'}.$$

Due to the product $\alpha\rho$, the two parameters are partially negatively correlated and specific statements about them require substantial data. Nevertheless, when pooling across multiple experiments, it appears that anhedonia is in fact related to a significant reduction in reward sensitivity but does not impact learning rate (Fig. 10.7B; Huys et al., 2013). Additional credence to this finding was given by the fact that a dopaminergic manipulation mostly affected the learning rate. This is consistent with a multiplicative change in the prediction error putatively reported by dopamine (Schultz et al., 1997). However, while an impact of anhedonia on the learning rate might have implied dopaminergic mechanisms, the origins of changes to reward sensitivity in depression remains uncertain (Treadway and Zald, 2011; Huys et al., 2015a).

The ability to learn from rewards is also thought to be affected in schizophrenia. The prominent involvement of dopamine suggested that this impairment may either arise through an impairment of striatal reward learning mechanisms, or alternatively also through impairment of prefrontal working memory mechanisms where dopamine also plays a key role (Durstewitz and Seamans, 2008). Collins et al. (2014) exploited a standard operant conditioning task which is, nevertheless, sensitive to both working memory and striatal prediction error learning mechanisms: when subjects are presented with increasing numbers of stimuli to learn about concurrently, a slowing of learning is observed (Fig. 10.7C). This pattern is not well accounted for by a simple change in learning rate and instead requires a working memory component to be postulated (Fig. 10.7D and E). Specifically, they consider a combination of two learners. The first is the reward learning module and is as in Eq. (10.11). The second, the working memory module, has a learning rate α set to 1. This means that the resulting $\mathcal{Q}_{wm}$ values store the previous event, and discard anything before that. After the choice, the $\mathcal{Q}_{wm}$ values are decayed to mimic forgetting. Strikingly, the impairment seen in schizophrenia was due mostly to the working memory component, rather than to the reward learning component.

10.4.2 Pavlovian Influences

We next turn to the distinction between two types of values: Pavlovian values of state $\mathcal{V}(s)$ and instrumental or operant values of state−action pairs $\mathcal{Q}(s, a)$. The former designate desirable states, but imply a policy or behavioral preference only via additional mechanisms, for instance, evolutionarily preprogrammed approach responses to appetitive states (Dayan et al., 2006). In contrast, the $\mathcal{Q}$ values measure the goodness of actions and hence can theoretically be used directly to motivate arbitrarily specific behaviors. There is a rich literature distinguishing these (see Dayan and Berridge, 2014; Huys et al., 2014 for reviews).

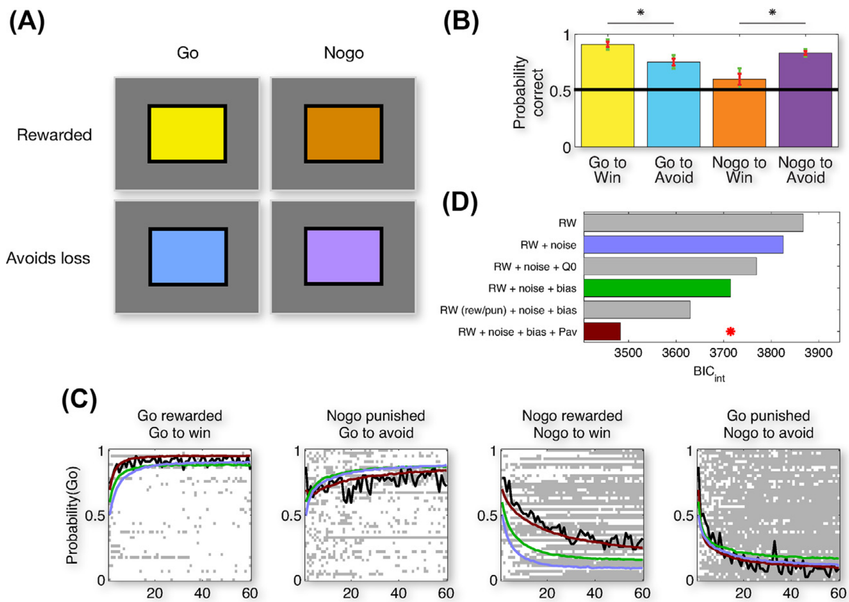

FIGURE 10.8 Pavlovian and instrumental components of choice. (A) Subjects were presented with one of four stimuli on each trial. For the yellow stimulus, go responses were rewarded and nogo not rewarded. For the orange stimulus, nogo responses were rewarded and go not rewarded. Similarly, for the blue stimulus go responses led to avoidance of a loss, while nogo responses led to avoidance of the loss for purple stimuli. (B) Overall pattern of results: performance is impaired when go and loss are paired, and when nogo and rewards are paired. (C) Learning curves. The background shows individual choices (go white, nogo gray) for each participant; *black lines* show averages over subjects; and *colored lines* are data generated from different models. (D) Model comparison, with the most parsimonious model having the lowest score (indicated with a *red star*). *BIC*, Bayesian Information Criterion; *Pav*: Pavlovian influence; *RW*: Rescorla–Wagner. *From Guitart-Masip, M., Huys, Q.J.M., Fuentemilla, L., Dayan, P., Duzel, E., Dolan, R.J., 2012. Go and no-go learning in reward and punishment: interactions between affect and effect. Neuroimage 62 (1), 154–166.*

Fig. 10.8A shows a very simple task that shows these components concurrently at work during learning in humans; when subjects have to go and are rewarded, or when they have to withhold going and are in a punishment context, they perform well, whereas performing go responses to avoid losses or nogo responses for reward is far more difficult (Fig. 10.8B). Looking at the learning curves (Fig. 10.8C), it appears clear that learning is slower in the two difficult scenarios. A simple model (blue) that only incorporates instrumental learning of stimulus–action values cannot account for this pattern. Incorporating a bias toward or away from performing go responses also fails to capture the data (green lines). It is only when a second, Pavlovian, learning mechanism is added to the instrumental learner that the performance across the four contexts can be matched, and then does so in sufficiently great detail to merit the increase in complexity (Fig. 10.8D). This Pavlovian influence simply promotes the active go choice in proportion to the average reward experienced for each stimulus,

$$\mathcal{V}_{t+1}(s) = \mathcal{V}_t(s) + \alpha(\rho r_t - \mathcal{V}_t(s))$$

$$w(a,s) = \begin{cases} \mathcal{Q}(a,s) + \in \mathcal{V}(s) & \text{if } a \text{ is go action} \\ \mathcal{Q}(a,s) & \text{else} \end{cases}$$

$$p(a_t|s_t) = \frac{\exp(w(a_t, s_t))}{\sum_{a'} \exp\left(w\left(a', s_t\right)\right)}$$

that is, when the stimulus leads to rewards, go is promoted, and when the stimulus tends to lead to losses, go is inhibited proportionally to the value of the stimulus. This is another instance where each individual appears to be influenced by multiple learning systems akin to Fig. 10.6E.

Though not examined with this particular task, the influence of Pavlovian stimulus-bound values on instrumental choices has been found to be aberrant in a variety of conditions ranging from alcoholism to depression. In alcoholism, for instance, Pavlovian influences are stronger, and the extent to which this involves the ventral striatum appears to predict relapse after detoxification (Garbusow et al., 2016).

10.4.3 Model-Based and Model-Free Decision-Making

A third example concerns the distinction between model-based and model-free decision-making. In model-based decision-making, the agent is assumed to know the consequences of actions and knows where rewards are located. This implies knowledge of transition matrices $\mathcal{T}$ and reward functions $\mathcal{R}$. At choice time, evaluations of different behavioral options are performed by searching the tree defined by $\mathcal{T}, \mathcal{R}$ (Daw et al., 2005; though see Daw and Dayan, 2014). In model-free decision-making the values $\mathcal{V}$ are accumulated over time through experience. At choice time, no further computation is required. The two types of decision-making thus trade computational costs for experiential costs. Daw et al. (2011) designed a task to measure the trade-off between the two types of learning within an individual.

Motivated by the suggestion that addictive and compulsive disorders might involve a shift from model-based toward model-free decision-making (Robbins et al., 2012), this task has since been examined extensively, with some supporting (Voon et al., 2015; Gillan et al., 2016), but also complicating evidence (Nebe et al., 2016). The difficulties stem particularly from the fact that the model-free component appears both poorly measured and unresponsive to any intervention (cf., Huys et al., 2016).

10.4.4 Complex Planning

We finally turn to a fourth example that uses RL techniques to examine how more complex planning tasks are solved (Huys et al., 2012, 2015c). The motivation for doing so is that many daily tasks involve planning problems that are extremely complex and easily overwhelm even powerful computers. They therefore cannot be solved fully, but mostly be approximated and simplified. Fig. 10.9A and B shows an example task that has to be solved by planning, but which is difficult. Fig. 10.9C and E show two possible strategies to approximate the task. The first, pruning, involves reflexively stopping the consideration of a plan if the plan requires transitioning through a salient loss (here, −70 points; cf. panel B). This means that large gains hiding behind the large losses are also missed. Indeed, subjects nearly never chose to transition through the path involving a large loss when there was another equally good path (Fig. 10.9D). Strikingly, when comparing the inferred tendency to stop thoughts at salient loss points, this effect appeared nearly independent of the size of the salient loss (Fig. 10.9E). If pruning were an adaptive response to the large loss, then this should have varied with loss size. This instead suggests a very simple, reflexive reaction to stop thoughts when salient losses are encountered. Further models examined how subjects subdivided the task (Fig. 10.9F). Strikingly, they subdivided the task in a manner that nearly optimally reduced the computational load (Fig. 10.9G).

10.5 DISCUSSION

Learning and decision-making are closely related facets of human affect and cognition. RL and dynamic programming provide principled approaches, which have been briefly reviewed here. This was followed by a brief, tutorial-like overview over how to fit such models to actual data. A point worth emphasizing is the importance of validating the model and of combining formal model comparison with informal comparisons of data generated from the model with the real data. Finally, the chapter covered a few prominent applications of the theory to psychiatric or neuroscience questions.

Taking a step back, one can ask what paths decision−theoretic accounts provide for psychiatric dysfunctions. One categorization is into three such paths (Huys et al., 2015b):

- Solving the wrong problem. This features the use of the wrong model of the world: either maximizing the wrong reward function (for instance, judging a short-term drug reward more important than long-term financial stability), or utilizing the wrong predictions about action consequences (wrongly believing that one becomes more socially adept when high), or interpreting events wrongly due to errors in the likelihood.

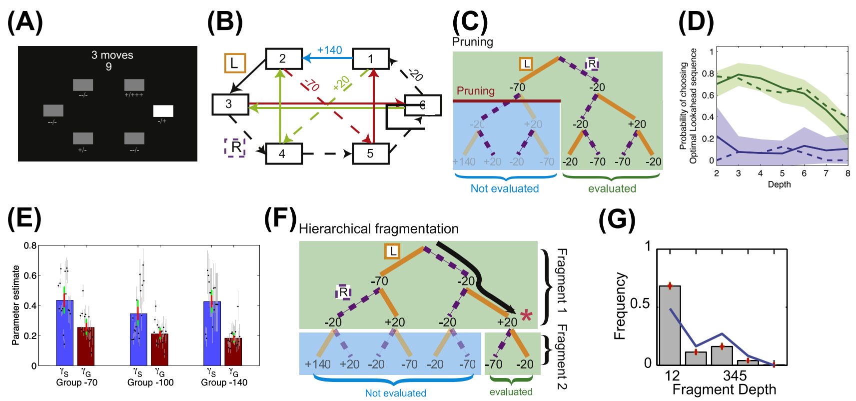

FIGURE 10.9 Task and approximations. (A) Subjects were shown six boxes. The randomly chosen starting location was indicated by the bright box and the number of moves to plan by the number at the top. Subjects were given time to plan, and then had to enter the entire planned sequence in terms of left/right button presses before seeing the chosen sequence and the rewards earned. (B) The task consisted of a maze, and subjects were placed in one of the six boxes at the beginning of each trial. They planned how to traverse the maze such as to maximize the sum of deterministic outcomes earned along the path. Each state had two successor states, which could be reached deterministically by right or left button presses. (C) Decision tree starting from state 3 and for a depth of 3 moves to plan. Pruning involves cutting off branches of the tree. A simple pruning strategy is to avoid transitions through large losses. In this particular setup with −70 as large losses, this would lead to the even larger gains being forfeited. (D) The lines show the fraction of optimal paths chosen for each depth of problem. In this version of the task, there were always two optimal paths: one through a salient loss (*blue line*), the other avoiding the salient loss (*green line*). When given the choice, subjects thus nearly deterministically avoided transitions through the large loss even when this had no impact on the outcome. (E) A computational measure of the probability of stopping the evaluation of a tree at a salient loss (blue) and at other points (red) for three groups with different salient losses of −70, −100, and −140. Strikingly, the stopping probabilities are barely different, suggesting that the inhibition of thoughts is reflexive rather than adaptively goal-directed itself. (F) Hierarchical decomposition. The complexity of the problem can be drastically reduced by approximating it with a subdivision of the task into smaller problems that are composed greedily. Here, for instance, first solving the depth-2 tree, and then solving whichever depth-1 tree this leads to. (G) The *blue line* shows the distribution of thought fragment lengths that would maximally reduce computational load without affecting performance. The *gray lines* are inferred from the data and show a close match, suggesting that individuals spontaneously near-optimally subdivided the task to minimize computational costs. *From Huys, Q.J.M., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., Roiser, J.P., 2012. Bonsai trees in your head: how the Pavlovian system sculpts goal-directed choices by pruning decision trees. PLoS Comput. Biol. 8 (3), e1002410; Huys, Q.J.M., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S.J., Dayan, P., Roiser, J.P., 2015c. Interplay of approximate planning strategies. Proc. Natl. Acad. Sci. U.S.A. 112 (10), 3098–3103.*

- Solving the correct problem, but poorly or wrongly. As most decision problems are too hard to solve, some measure of approximation and error will naturally occur. The examples in the previous section show that these features are actively being investigated.
- Solving the correct problem, correctly, but based on poor experience. Trauma and stress are strongly associated with psychiatric ill-health. Behavior following traumatic exposure may well represent the "correct" solution even though it impairs well-being.

Finally, it should be mentioned that these techniques may well be useful in combination with other techniques. For instance, the extraction of meaningful parameters in a generative model may provide a very accurate and informationally efficient summary of complex, high-dimensional data. As such, these models can function preprocessing to reduce the dimensionality of data before applying other analyses (Wiecki et al., 2015, 2016; Huys et al., 2016).

References

Bellman, R.E., 1957. Dynamic Programming. Princeton University Press.

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-Dynamic Programming. Athena Scientific.

Collins, A.G.E., Brown, J.K., Gold, J.M., Waltz, J.A., Frank, M.J., 2014. Working memory contributions to reinforcement learning impairments in schizophrenia. J. Neurosci. 34 (41), 13747−13756.

Daw, N., 2009. Trial-by-trial data analysis using computational models. In: Delgado, M.R., Phelps, E.A., Robbins, T.W. (Eds.), Decision Making, Affect, and Learning: Attention and Performance XXIII. OUP.

Daw, N.D., Dayan, P., 2014. The algorithmic anatomy of model-based evaluation. Philos. Trans. R. Soc. Lond. B Biol. Sci. 369 (1655).

Daw, N.D., Gershman, S.J., Seymour, B., Dayan, P., Dolan, R.J., 2011. Model-based influences on humans' choices and striatal prediction errors. Neuron 69 (6), 1204−1215.

Daw, N.D., Niv, Y., Dayan, P., 2005. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nat. Neurosci. 8 (12), 1704−1711.

Dayan, P., Berridge, K.C., 2014. Model-based and model-free pavlovian reward learning: revaluation, revision, and revelation. Cogn. Affect. Behav. Neurosci. 14 (2), 473−492.

Dayan, P., Niv, Y., Seymour, B., Daw, N.D., 2006. The misbehavior of value and the discipline of the will. Neural Netw. 19 (8), 1153−1160.

Durstewitz, D., Seamans, J.K., 2008. The dual-state theory of prefrontal cortex dopamine function with relevance to catechol-o-methyltransferase genotypes and schizophrenia. Biol. Psychiatry 64 (9), 739−749.

Garbusow, M., Schad, D.J., Sebold, M., Friedel, E., Bernhardt, N., Koch, S.P., Steinacher, B., Kathmann, N., Geurts, D.E.M., Sommer, C., Müller, D.K., Nebe, S., Paul, S., Wittchen, H.-U., Zimmermann, U.S., Walter, H., Smolka, M.N., Sterzer, P., Rapp, M.A., Huys, Q.J.M., Schlagenhauf, F., Heinz, A., 2016. Pavlovian-to-instrumental transfer effects in the nucleus accumbens relate to relapse in alcohol dependence. Addict. Biol. 21 (3), 719−731.

Gillan, C.M., Kosinski, M., Whelan, R., Phelps, E.A., Daw, N.D., 2016. Characterizing a psychiatric symptom dimension related to deficits in goal-directed control. Elife 5.

Guitart-Masip, M., Huys, Q.J.M., Fuentemilla, L., Dayan, P., Duzel, E., Dolan, R.J., 2012. Go and no-go learning in reward and punishment: interactions between affect and effect. Neuroimage 62 (1), 154−166.
Huys, Q.J.M., Dayan, P., Daw, 2015a. Depression: a decision-theoretic account. Ann. Rev. Neurosci. 38, 1−23.
Huys, Q.J.M., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., Roiser, J.P., 2012. Bonsai trees in your head: how the Pavlovian system sculpts goal-directed choices by pruning decision trees. PLoS Comput. Biol. 8 (3), e1002410.
Huys, Q.J.M., Guitart-Masip, M., Dolan, R.J., Dayan, P., 2015b. Decision-theoretic psychiatry. Clin. Psychol. Sci. 3 (3), 400−421.
Huys, Q.J.M., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S.J., Dayan, P., Roiser, J.P., 2015c. Interplay of approximate planning strategies. Proc. Natl. Acad. Sci. U.S.A. 112 (10), 3098−3103.
Huys, Q.J.M., Maia, T.V., Frank, M.J., 2016. Computational psychiatry as a bridge from neuroscience to clinical applications. Nat. Neurosci. 19 (3), 404−413.
Huys, Q.J.M., Pizzagalli, D.A., Bogdan, R., Dayan, P., 2013. Mapping anhedonia onto reinforcement learning: a behavioural meta-analysis. Biol. Mood Anxiety Disord. 3 (1), 12.
Huys, Q.J.M., Tobler, P.N., Hasler, G., Flagel, S.B., 2014. The role of learning-related dopamine signals in addiction vulnerability. Prog. Brain Res. 211, 31−77.
Kass, R., Raftery, A., 1995. Bayes factors. J. Am. Stat. Assoc. 90 (430).
Kendler, K.S., Karkowski, L.M., Prescott, C.A., 1999. Causal relationship between stressful life events and the onset of major depression. Am. J. Psychiatry 156, 837−841.
Nebe, S., Kroemer, N.B., Schad, D.J., Bernhardt, N., Sebold, M., Müller, D.K., Scholl, L., Kuitunen-Paul, S., Heinz, A., Rapp, M., Huys, Q.J., Smolka, M.N., 2017. No association of goal-directed and habitual control with alcohol consumption in young adults, Addict. Biol. http://dx.doi.org/10.1111/adb.12490 (epub ahead of publication).
Pizzagalli, D.A., Jahn, A.L., O'Shea, J.P., 2005. Toward an objective characterization of an anhedonic phenotype: a signal-detection approach. Biol. Psychiatry 57 (4), 319−327.
Rescorla, R., Wagner, A., 1972. A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. Class. Cond. Curr. Res. Theory 64−99.
Robbins, T.W., Gillan, C.M., Smith, D.G., de Wit, S., Ersche, K.D., 2012. Neurocognitive endophenotypes of impulsivity and compulsivity: towards dimensional psychiatry. Trends Cogn. Sci. 16 (1), 81−91.
Schultz, W., Dayan, P., Montague, P.R., 1997. A neural substrate of prediction and reward. Science 275 (5306), 1593−1599.
Stephan, K.E., Penny, W.D., Daunizeau, J., Moran, R.J., Friston, K.J., 2009. Bayesian model selection for group studies. Neuroimage 46 (4), 1004−1017.
Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.
Treadway, M.T., Zald, D.H., 2011. Reconsidering anhedonia in depression: lessons from translational neuroscience. Neurosci. Biobehav. Rev. 35 (3), 537−555.
Voon, V., Derbyshire, K., Rück, C., Irvine, M.A., Worbe, Y., Enander, J., Schreiber, L.R.N., Gillan, C., Fineberg, N.A., Sahakian, B.J., Robbins, T.W., Harrison, N.A., Wood, J., Daw, N.D., Dayan, P., Grant, J.E., Bullmore, E.T., 2015. Disorders of compulsivity: a common bias towards learning habits. Mol. Psychiatry 20 (3), 345−352.
Watkins, C., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3), 279−292.
Wiecki, T.V., Poland, J., Frank, M.J., 2015. Model-based cognitive neuroscience approaches to computational psychiatry clustering and classification. Clin. Psychol. Sci. 3 (3), 378−399.
Wiecki, T.V., Antoniades, C.A., Stevenson, A., Kennard, C., Borowsky, B., Owen, G., Leavitt, B., Roos, R., Durr, A., Tabrizi, S.J., Frank, M.J., 2016. A computational cognitive biomarker for early-stage huntington's disease. PLoS One 11 (2), e0148409.