11
Can quantum systems learn? Quantum updating

Learning is a critical aspect of any intelligent cognitive system. How can this be done within a QIP approach? This is a relatively new field, but some progress has already been achieved (Ivancevic & Ivancevic, 2010). There are at least three ways to accomplish learning using quantum principles. One way is to update the agent's belief state based on experience (Schack *et al.*, 2001), as done in Bayesian learning models (Griffiths *et al.*, 2008). A second way is to update the weights in a unitary matrix using gradient descent of an error function (Zak & Williams, 1998), as done in connectionist learning models (Rumelhart & McClelland, 1986). A third way is to update the amplitudes assigned to control U gate actions based on rewards and punishments (Dong *et al.*, 2010), as done with reinforcement learning algorithms (Sutton & Barto, 1998). This chapter reviews all three approaches.

11.1 Quantum state updating based on experience

For the first type of quantum learning model, consider how to update an agent's belief state based on experience. In Chapter 4 we presented a quantum model for probability judgments, and in that chapter the initial belief state denoted $|\psi\rangle$ was given or assumed to be already known in advance – when new facts were presented, inferences were made from the known state $|\psi\rangle$ using Lüder's rule. However, where does this initial state $|\psi\rangle$ come from? Now we examine how this initial state $|\psi\rangle$ can be learned or estimated from experience. Principles borrowed from quantum state tomography can be used to model the estimation of a quantum state (Schack *et al.*, 2001).

11.1.1 Quantum Bayes nets

Recall from Chapter 4 that a state vector can be used to represent a person's beliefs about combinations of values for a set of features (features are also called variables). Suppose there are four features labeled u, v, w, x, and, for simplicity, suppose each feature is binary valued with values $\{0, 1\}$. To be more concrete, suppose variable u represents the presence ($u = 1$) or absence ($u = 0$) of a genetic disposition for a neural disease. The latter can influence v which represents

the presence or absence of a malfunctioning neural system. The variable w represents the presence or absence of an environmental stress, and w together with v influence the variable x, which represents the presence or absence of a psychopathological state. All combinations of the binary values for the four features form a total set of 16 unique event patterns.

For now we assume that all four features are compatible, in which case we can represent the belief state for these four binary variables by a vector $|\psi\rangle$ within an $n = 16$-dimensional Hilbert space spanned by 16 orthonormal basis vectors:

$$|\psi\rangle = \sum_{ijkl} \psi_{ijkl} \cdot |u_i v_j w_k x_l\rangle. \tag{11.1}$$

Each basis vector $|u_i v_j w_k x_l\rangle$ represents a combination of the binary values from the four variables. In particular, $|u_0 v_1 w_0 x_1\rangle$ represents the occurrence of the event that $u = 0$ and $v = 1$ and $w = 0$ and $x = 1$ (i.e., the event that there is no genetic predisposition, but there is a presence of a malfunctioning neural system, and there is no environmental stress, but psychopathological state is present). The coordinate ψ_{ijkl} assigned to a basis vector determines the probability amplitude for a combination of feature values. For example, ψ_{0101} represents the probability amplitude assigned to the combination of feature values $u = 0$, $v = 1, w = 0, x = 1$ (i.e., the probability amplitude that there is no genetic predisposition, but there is a presence of a malfunctioning neural system, and there is no environmental stress, but psychopathological state is present). The 16×1 matrix $\psi = [\psi_{ijkl}]$ contains all 16 coordinates for all 16 probability amplitudes ψ_{ijkl}, $i, j, k, l \in \{0, 1\}$, assigned to the 16 basis vectors.

The problem of learning a quantum state from experience is analogous to learning the probabilities in a Bayesian causal network (Pearl, 1988). Quantum Bayes nets can be used to represent dependencies among variables that define the basis for a quantum state (Tucci, 1995).[1] Consider, for example, the acyclic "causal" network shown in Figure 11.1. This represents one possible causal model for describing the causal relations among variables u, v, w, x.

Using Figure 11.1 as the basis for our model, we can assign probability amplitudes to ψ in this network using quantum principles analogous to principles used in Bayesian networks (Tucci, 1995). The basic idea is that any classic Bayes net can be extended to a quantum Bayes net by replacing the probabilities in the classic model with amplitudes in the quantum model. For our example in Figure 11.1, we can set

$$\psi_{ijkl} = \psi(u = i) \cdot \psi(v = j|u = i) \cdot \psi(w = k) \cdot \psi(x = l|v = j, w = k). \tag{11.2}$$

In the above expression we define $\psi(u = i)$ as the probability amplitude assigned to value i of variable u with the constraint $\sum_i |\psi(u = i)|^2 = 1$; we define $\psi(v = j|u = i)$ as the conditional probability amplitude of value j for variable v given that variable u is observed to have value i, with the constraint

[1] Also see Tucci (1997). The quantum Bayes nets developed by Tucci are acyclic. However, La Mura and Swiatczak (2007) extended the quantum model to include cyclic networks.

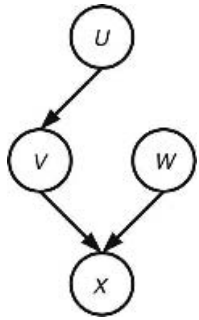

Figure 11.1 Example of a causal network in which variable u influences variable v, and both variables v and w influence variable x.

$\sum_j |\psi\,(v = j|u = i)|^2 = 1$ for each i; we define $\psi(w = k)$ as the probability amplitude for value k of variable w, with the constraint $\sum_k |\psi(w = k)|^2 = 1$; and we define $\psi(x = l|v = j, w = k)$ as the conditional probability amplitude that variable x has value l given that variable v is observed to have value j and variable w is observed to have value k, with the constraint $\sum_l |\psi(x = l|v = j, w = k)|^2 = 1$ for each pair of values for j, k. These constraints on the probability amplitudes guarantee that $\left|\left|\psi^2\right|\right| = 1$. As shown below, the conditional amplitudes of Eq. (11.2) generate probabilities that are consistent with the results derived by applying Lüder's rule to the original amplitudes in Eq. (11.1) after observing the conditioning events.

The standard quantum probability rules are applied to this state to generate all the required marginal and conditional event probabilities needed to make inferences from the network shown in Figure 11.1. Consider, for example, the conditional probability $p(x = 1|v = j, w = k)$. Starting from basic quantum principles, we define a projector for observing the event $v = j$ as $P_{v=j}$, which corresponds to a 16×16 indicator matrix that picks out the amplitudes corresponding to the event $v = j$ from ψ in Eq. (11.1). We also define a projector for event $w = k$ as $P_{w=k}$, which corresponds to another 16×16 indicator matrix that picks out the amplitudes corresponding to event $w = k$ from ψ in Eq. (11.1). Recall that, according to Lüder's rule, if we observe the events $v = j$ and $w = k$, then the state changes from the initial state ψ to the revised state:

$$\psi_{v=j,w=k} = \frac{P_{v=j} \cdot P_{w=k} \cdot \psi}{\|P_{v=j} \cdot P_{w=k} \cdot \psi\|} = \frac{P_{v=j} \cdot P_{w=k}\psi}{\sqrt{p(v = j, w = k)}},$$

and note that

$$\begin{aligned} p(v = j, w = k) &= \|P_{v=j} \cdot P_{w=k} \cdot \psi\|^2 \\ &= \sum_i \sum_l |\psi(u = i)|^2 \cdot |\psi(v = j|u = i)|^2 \cdot |\psi(w = k)|^2 \\ &\quad \cdot |\psi(x = l|v = j, w = k)|^2 \end{aligned}$$

$$= \sum_i |\psi(u=i)|^2 \cdot |\psi(v=j|u=i)|^2 \cdot |\psi(w=k)|^2$$
$$\cdot \sum_l |\psi(x=l|v=j,w=k)|^2$$
$$= \sum_i |\psi(u=i)|^2 \cdot |\psi(v=j|u=i)|^2 \cdot |\psi(w=k)|^2 \cdot 1,$$

so that the probability of observing $x=1$ after observing the other two events $v=j$ and $w=k$ equals

$$p(x=1|v=j,w=k) = \left\| P_{x=1} \cdot \psi_{v=j,w=k} \right\|^2$$
$$= \frac{\|P_{x=1} \cdot P_{v=j} \cdot P_{w=k} \cdot \psi\|^2}{\|P_{v=j} \cdot P_{w=k} \cdot \psi\|^2}$$
$$= \frac{\|P_{x=1} \cdot P_{v=j} \cdot P_{w=k} \cdot \psi\|^2}{p(v=j,w=k)}$$
$$= \frac{\left(\sum_i |\psi(u=i)|^2 \cdot |\psi(v=j|u=i)|^2 \cdot |\psi(w=k)|^2\right) \cdot |\psi(x=1|v=j,w=k)|^2}{p(v=j,w=k)}$$
$$= \frac{p\,(v=j,w=k) \cdot |\psi(x=1|v=j,w=k)|^2}{p(v=j,w=k)} = |\psi(x=1|v=j,w=k)|^2 .$$

The last line proves that $|\psi(x=1|v=j,w=k)|^2$ equals the same probability as obtained by applying Lüder's rule to Eq. (11.1) after observing the events $v=j$ and $w=k$. As another example, suppose we learn that $u=1$; and consider the conditional probability $p\,(x=1|u=1)$. Then this conditional probability equals

$$p\,(x=1|u=1) = \frac{\|P_{x=1} \cdot P_{u=1} \cdot \psi\|^2}{\|P_{u=1} \cdot \psi\|^2} = \frac{\sum_j \sum_k \left|\psi_{1jk1}\right|^2}{|\psi\,(u=1)|^2}$$
$$= \sum_j \sum_k |\psi\,(v=j|u=1)|^2 \cdot |\psi\,(w=k)|^2 \cdot |\psi\,(x=1|v=j,w=k)|^2 .$$

Alternatively, we can compute the probability of $u=1$ given $x=1$:

$$p(u=1|x=1) = \frac{\|P_{u=1} \cdot P_{x=1} \cdot \psi\|^2}{\|P_{x=1} \cdot \psi\|^2}$$
$$= \frac{\|P_{u=1} \cdot P_{x=1} \cdot \psi\|^2}{\|(P_{u=0} + P_{u=1}) \cdot P_{x=1} \cdot \psi\|^2}$$

$$= \frac{\|P_{u=1} \cdot P_{x=1} \cdot \psi\rangle\|^2}{\|P_{u=0} \cdot P_{x=1} \cdot \psi\|^2 + \|P_{u=1} \cdot P_{x=1} \cdot \psi\|^2}$$

$$= \frac{\|P_{u=1} \cdot \psi\|^2 \|P_{x=1} \cdot \psi_{u=1}\|^2}{\|P_{u=0} \cdot \psi\|^2 \|P_{x=1} \cdot \psi_{u=0}\|^2 + \|P_{u=1} \cdot \psi\|^2 \|P_{x=1} \cdot \psi_{u=1}\|^2}$$

$$= \frac{p\,(u=1) \cdot p\,(x=1|u=1)}{\sum_i p\,(u=i) \cdot p\,(x=1|u=i)}.$$

As can be seen from the above examples, inference is performed with squared amplitudes in a manner directly analogous to Bayesian causal networks used in psychological theories of causal reasoning (Griffiths *et al.*, 2008).

Why derive probabilities from amplitudes if squared magnitudes of amplitudes simply reproduce the same results as using probabilities directly? The answer is that amplitudes become critical as soon as we introduce a new set of incompatible features. For example, suppose the basis $|u_i v_j w_k x_l\rangle$ is used to evaluate the probabilities generated by the variables in Figure 11.1 from an impersonal perspective of an institution (e.g., an expert providing opinions for the medical insurance industry) and the coordinates for this basis are represented by the amplitudes in ψ. Alternatively, suppose another orthonormal basis $\left|u'_i v'_j w'_k x'_l\right\rangle$ is used to evaluate the same events shown in Figure 11.1, but now from a personal perspective concerning the specific life of a single individual (e.g., providing judgments about a highly familiar patient) and the coordinates for this basis are represented by the amplitudes in $\phi = U \cdot \psi$. Suppose an expert is asked to evaluate whether a "genetic disposition is present" from the impersonal perspective of an insurance industry ($u = 1$), given that "psychopathological behavior is present" from the personal perspective of a single individual ($x' = 1$); or in other words, $p(u = 1|x' = 1)$. According to a quantum inference model, this probability is obtained by the following three steps. First, the personal basis $\left|u'_i v'_j w'_k x'_l\right\rangle$ is used to represent the belief state $|\psi\rangle$, which assigns the amplitudes according to ϕ. This belief state is updated on the basis of the fact that the personal event $x' = 1$ is observed by picking out the coordinates in ϕ corresponding to the event $x' = 1$ using the projector $P_{x'=1}$ and normalizing the result to produce the conditional amplitudes

$$\phi_{x'=1} = \frac{P_{x'=1}\phi}{\sqrt{p(x'=1)}}.$$

Then the basis is changed to $|u_i v_j w_k x_l\rangle$ in order to evaluate the impersonal insurance industry perspective, and this basis reassigns amplitudes according to the transformation

$$\psi_{x'=1} = U^\dagger \cdot \phi_{x'=1}.$$

Finally, the probability of $u = 1$ from the impersonal perspective given that $x' = 1$ is observed from the personal perspective equals

$$p\,(u=1|x'=1) = \|P_{u=1}\psi_{x'=1}\|^2.$$

In sum, by using amplitudes and unitary transformations, quantum theory provides a simple way to represent two different causal belief systems – one from a personal perspective and one from an impersonal perspective, and it also provides a simple and efficient way to evaluate sequences of evidence obtained from these different perspectives. Changes in the basis used to evaluate events produce non-commutative events that lead to order effects (see Chapters 3 and 4) and interference effects (see Chapter 9) and entanglement effects (see Chapters 5 and 7) that are difficult to explain using standard (Kolmogorov and Bayesian) probability theory.

11.1.2 Updating amplitudes based on experience

The amplitudes for a causal network are learned on the basis of observations generated by the causal network. Suppose we need to learn the probability amplitude vector

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \psi\left(x = 0 | v = j, w = k\right) \\ \psi\left(x = 1 | v = j, w = k\right) \end{bmatrix},$$

with the constraint that $\|\alpha\|^2 = 1$. The set of all possible α forms a sphere with unit radius called the Bloch sphere (Nielsen and Chuang, 2000). Our prior distribution places probabilities on each point on this Bloch sphere. The prior probability density assigned to each point on the Bloch sphere is denoted $p(\alpha)$.

First, suppose α is known and we obtain N identical and independent observations on variable x (conditioned on observed values for $u = j, v = k$). The results are summarized by the pair of data points $R = (n_0, n_1)$, where n_0 is the number of times we observe $x = 0$ and n_1 is the number of times we observe $x = 1$, so that $N = n_0 + n_1$. According to the quantum model, the probability of obtaining $R = (n_0, n_1)$ given the state α equals the binomial distribution

$$p(n_0, n_1|\alpha) = \binom{N}{n_1} \cdot \left(|\alpha_1|^2\right)^{n_1} \cdot \left(|\alpha_0|^2\right)^{n_0}.$$

The posterior probability of α given the observations in R is obtained by Bayes' rule (Schack *et al.*, 2001):

$$p(\alpha|n_0, n_1) = \frac{p(\alpha) \cdot p(R|\alpha)}{\int p(R|\alpha)p(\alpha)\ \mathrm{d}\alpha}.$$

One commonly used prior is defined by uniform sphere integration. In this case the updating rule becomes (Jones, 1991)

$$p(\alpha|n_0, n_1) = (N + 1) \cdot \binom{N}{n_1} \cdot \left(|\alpha_1|^2\right)^{n_1} \cdot \left(|\alpha_0|^2\right)^{n_0}$$

(see Equation 23 in Jones (1991)).

Alternatively, we could assume a much more restrictive prior. In particular, we could assume that α is initially restricted to real values so that $\alpha_1 =$

$= \sqrt{p\left(x = 1 | v = j, w = k\right)}$. Then we can use a beta distribution $B(a+1, b+1)$ with a, b positive integers and $M = a + b$ to define the prior (Kruschke, 2010):

$$p\left(\alpha_1^2\right) = (M+1) \cdot \binom{M}{a} \cdot \left(\alpha_1^2\right)^a \cdot \left(\alpha_0^2\right)^b.$$

Using this beta prior, the posterior distribution equals

$$p\left(\alpha_1^2 | n_0, n_1\right) = (N+M+1) \cdot \binom{N+M}{n_1+a} \alpha_1^{2(n_1+a)} \cdot \alpha_0^{2(n_0+b)}.$$

If we now wish to change variables from α_1^2 to α_1, then we have to use the change in rate $\left|\frac{\partial \alpha_1^2}{\partial \alpha_1}\right| = 2 \cdot \alpha_1$ to form the transformed density $p\left(\alpha_1 | R\right) = 2 \cdot p\left(\alpha_1^2 | R\right) \cdot \alpha_1$.

In summary, according to quantum probability theory, amplitudes are basic and probabilities are derived from amplitudes. Learning the state for a quantum network requires estimating the amplitudes for the nodes in the causal network from experience. In particular, for the causal network shown in Figure 11.1, learning involves estimating the amplitudes for $\psi\left(u = i\right)$, $\psi\left(v = j | u = i\right)$, $\psi\left(w = k\right)$, and $\psi\left(x = l | v = j, w = k\right)$ from experience.

Quantum theory also allows new causal belief networks to be formed by unitary transformations of the basis. However, this raises the following question: How does a person learn a unitary transformation that changes from one basis to the next? This issue is addressed in the next section.

11.2 Weight updating based on gradient descent learning

For a second type of quantum learning model, consider learning the "connection weights" u_{jk} in a unitary matrix U based on experience with pairs of input and output probability amplitude distributions, analogous to the learning of input–output activation patterns by a connectionist learning model. This idea has been used as the basis for quantum neural network learning models (Kouda *et al.*, 2005).[2]

The basic idea is illustrated in Figure 11.2 for the special case of a four-dimensional Hilbert space. Information is presented to a person (represented by S in the figure) and this information generates a belief state $|\psi\rangle$ which can be used to answer various kinds of questions. For example, S may contain information about the health of an individual. In the figure, this belief state is initially interpreted with respect to the four basis vectors C_1 to C_4. For example, C_1 to C_4 could represent four possible answers to questions about a person's health status when evaluated from an impersonal perspective of an institution such as an insurance company. This input basis assigns an amplitude distribution ψ to

[2] An alternative way to learn with quantum neural nets is to use a Grover (1997) amplitude amplification algorithm (Ventura & Martinez, 2000). An application of the Grover algorithm is presented in the next section on reinforcement learning.

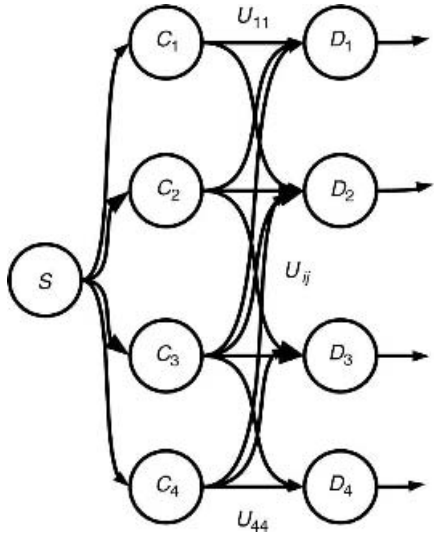

Figure 11.2 The inputs into C_1 to C_4 are connected to the outputs D_1 to D_4 by the connection weights u_{ij}.

the four events, so that the probability of observing an event corresponding to C_j equals $|\psi_j|^2$. Later this information is reinterpreted with respect to another set of four basis vectors D_1 to D_4. For example, D_1 to D_4 could represent four possible answers to questions about a person's health status when evaluated from a personal perspective about a single individual. The evaluator eventually learns a connection u_{jk} between each of four input basis vectors C_j and each of the four output basis vectors D_k to form a connection weight matrix U. These connection weights u_{jk} are eventually used to map the input amplitude distribution ψ across C_1 to C_4 into the output amplitude distribution ϕ across D_1 to D_4 by a simple linear combination rule $\phi_k = \sum_j u_{jk} \cdot \psi_j$. The final probability of observing an event corresponding to D_k is the squared magnitude $|\phi_k|^2$. Thus, the amplitudes are mapped in a simple linear manner but the observed event probabilities are related in a nonlinear manner.[3] Before presenting the learning model, we need to discuss more specifically the feedback information that is used for training in this learning task.

In general, a probability amplitude assigned to a basis vector is a complex number, $\phi_j = |\phi_j| \cdot e^{i\theta_j}$, with a magnitude $|\phi_j|$ and phase $e^{i\theta_j}$. The probability of the event pattern equals the squared magnitude $|\phi_j|^2$. A key assumption for this learning model is that a person can report a probability for each event pattern, which is determined from the squared magnitude of the amplitude assigned to its basis vector. This implies that an individual is aware of the

[3] A possible neural interpretation of the connection weight matrix is provided in Chapter 12. However, this transformation is closely related to Fourier transformations that are used to model features in the auditory and visual systems.

magnitude $|\phi_j|$ of an amplitude for a basis vector. However, we do not assume that a person is aware or able to report the phase information $e^{i\theta_j}$ about a probability amplitude. In quantum theory, only the probabilities (and thus the magnitude of the amplitude) are directly measurable, and the phase only has indirect effects that are expressed through the observed response probabilities.

When a piece of information is presented, the person initially represents the input with respect to the $|C_j\rangle$ basis and represents the information by an $N \times 1$ matrix $\psi = \left[\sqrt{p_j}\right]$, for $j = 1, \ldots, N$, where p_j equals the probability assigned to basis vector $|C_i\rangle$, with $\psi^\dagger \psi = 1$. When the person views this same information from the perspective of the $|D_j\rangle$ basis, the person experiences an $N \times 1$ matrix $\bar{\phi}$ containing the absolute values $\bar{\phi} = \left[\left|\phi_j\right|\right]$. Therefore, we assume that the person experiences input–output pairs $\left(\psi, \bar{\phi}\right)$ across trials, and from these pairs the associative learning system gradually forms connections U that map ψ into ϕ. After training, the person is able to generate an appropriate output ϕ when given any new input ψ. Suppose the person experiences a sequence of input–output pairs $\left(\psi_t, \bar{\phi}_t\right)$ and there is a unitary matrix U such that $\phi_t = U \cdot \psi_t$. Define the estimate of the unitary transformation after t trials of experience as $\hat{U}_t$. The next predicted output generated by this estimate is denoted $\hat{\phi}_{t+1}$, which is an $N \times 1$ matrix containing only the magnitudes of the coordinates of the transformed input amplitudes $\hat{U}_t \cdot \psi_{t+1}$. The squared distance between the observed output and the prediction equals

$$
\begin{aligned}
F_t &= \left\| \bar{\phi}_t - \hat{\phi}_t \right\|^2 \\
&= \left\| \bar{\phi}_t \right\|^2 + \left\| \hat{\phi}_t \right\|^2 - \left(\bar{\phi}_t^\dagger \cdot \hat{\phi}_t + \hat{\phi}_t^\dagger \cdot \bar{\phi}_t \right) \\
&= 2 - \left(\bar{\phi}_t^\dagger \cdot \hat{\phi}_t + \hat{\phi}_t^\dagger \cdot \bar{\phi}_t \right)
\end{aligned}
$$

and $R_t = \left(\bar{\phi}_t^\dagger \cdot \hat{\phi}_{t+1} + \hat{\phi}_{t+1}^\dagger \cdot \bar{\phi}_t\right)$ is twice the cosine of the angle between the target and the prediction. A trial-by-trial learning algorithm is formed by updating the estimate of the unitary matrix in a direction that maximizes the increase in this cosine (Toronto & Ventura, 2006).[4]

Recall from Chapter 2 that any unitary matrix can be expressed as a matrix exponential of a Hamiltonian matrix, $U = e^{-iH}$, where $H^\dagger = H$ is an $N \times N$ Hermitian matrix with elements $h_{jk} = h_{kj}^*$. Hereafter, we will assume that H is real so that $h_{jk} = h_{kj}$ (but note that $U = e^{-iH}$ is still a complex matrix and the phase makes its critical impact here). Under the latter constraint, the learning

[4]The algorithm below is closely related to the earlier one developed by Toronto and Ventura (2006) with the following important differences: Toronto and Ventura did not form the unitary from a Hamiltonian and so their transformation is not required to be unitary; they did not restrict the target output to contain only the magnitude of an amplitude and instead they assumed that both phase and magnitude are provided as feedback; they used a batch rather than trial-by-trial updating algorithm.

algorithm can be re-expressed in terms of the gradient of R_t with respect to the real values in H:

$$\nabla_t = \frac{\partial R_t(H)}{\partial H} = \left[\frac{\partial R_t}{\partial h_{jk}}\right].$$

In the above definition of the gradient, ∇_t is an $N \times N$ matrix with element $\delta_{jk} = \frac{\partial R_t}{\partial h_{jk}}$ in row j for $j = 1, \ldots, N$ and column k for $k = 1, \ldots, j$ and we require $\delta_{kj} = \delta_{jk}$ so that H remains symmetric. (This gradient is computed by a numerical finite-difference method in the program provided in Appendix G). Then the learning algorithm can be described by

$$H_t = H_{t-1} + s \cdot \nabla_t \tag{11.3}$$

$$\hat{U}_t = \mathrm{e}^{-iH_t}$$

and finally $\bar{\phi}_t$ contains the absolute values of the coordinates in $\hat{U}_t \cdot \psi_t$. The learning rate parameter s is a real-valued scalar. One additional step can be added to this algorithm – the previous estimate of H_{t-1} is changed to the new estimate H_t after trial t if and only if $R(H_t) > R(H_{t-1})$. In other words, only improvements in performance are accepted.

11.2.1 Example application

To examine how well this learning model works, consider the problem of learning the 4×4 unitary matrix described in Chapter 4, which was used to explain the order effects on inference in a juror decision task. The unitary matrix used in Chapter 4 was formed from the following Hamiltonian:

$$H = \begin{bmatrix} \frac{h}{\sqrt{1+h^2}} + \frac{-\gamma}{\sqrt{2}} & \frac{1}{\sqrt{1+h^2}} & \frac{-\gamma}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{1+h^2}} & \frac{-h}{\sqrt{1+h^2}} + \frac{\gamma}{\sqrt{2}} & 0 & \frac{-\gamma}{\sqrt{2}} \\ \frac{-\gamma}{\sqrt{2}} & 0 & \frac{h}{\sqrt{1+h^2}} + \frac{\gamma}{\sqrt{2}} & \frac{1}{\sqrt{1+h^2}} \\ 0 & \frac{-\gamma}{\sqrt{2}} & \frac{1}{\sqrt{1+h^2}} & \frac{-h}{\sqrt{1+h^2}} + \frac{-\gamma}{\sqrt{2}} \end{bmatrix}. \tag{11.4}$$

The unitary matrix was then obtained by the matrix exponential $U = \mathrm{e}^{-i\left(\frac{\pi}{2}\right)H}$. Recall from Chapter 4 that this unitary matrix was used to describe the change in basis used to make inferences in a juror decision task when evidence was viewed from either a defense on a prosecutor perspective.

The following procedure was used to sample input distributions. Each input amplitude distribution ψ_t was formed by randomly sampling N coordinates from a uniform distribution between zero and one and then normalizing the amplitudes so that $\psi_t^\dagger \cdot \psi_t = 1$.

The initial estimate for the unitary matrix was set equal to an identity matrix, $\hat{U}_0 = I$. This produced the first predicted output amplitude distribution for the first trial equal to $\hat{\phi}_1 = I \cdot \psi_1$. This initial prediction was then compared with the first target produced by the absolute values of the coordinates of $\phi_1 = U \cdot \psi_1 = \mathrm{e}^{-i\left(\frac{\pi}{2}\right)H} \cdot \psi_1$ with H defined in Eq. (11.4). The first prediction and

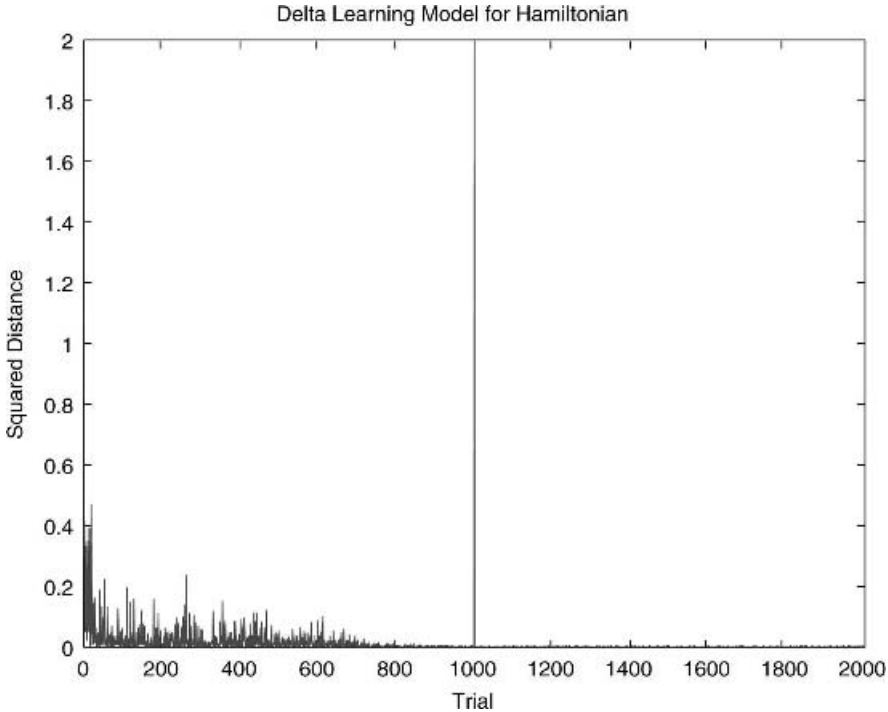

Figure 11.3 Squared distance between target and predicted output as a function training trial. The vertical line indicates where learning stopped and transfer testing began.

target were used to generate the initial squared distance $\left\|\bar{\phi}_1 - \hat{\phi}_1\right\|^2$, and the latter squared distance was used to start the learning algorithm.

Thereafter, the learning algorithm in Eq. (11.3) was applied for 1000 trials with $s = 0.25$. After 1000 trials, the step was set to zero to stop learning and the unitary matrix trained up to trial 1000 was then used to make predictions for the next 1000 trials. (The Matlab computer program used to calculate the results is provided in Appendix G.)

Figure 11.3 illustrates the efficiency of this learning model for the parameter values $h = 0.5167$ and $\gamma = 2.3109$ for Eq. (11.4). The squared distance rapidly drops from a starting value of 2 to a much lower value near 0.10 and it is eventually driven to zero. This result is typical for hundreds of simulations using these same parameters. Very similar results were also obtained by varying the parameters for the Hamiltonian and varying the step size s.

It is interesting to compare the target transition matrix T generated by the target U matrix with the estimated transition matrix $\hat{T}$ generated by the final estimate $\hat{U}$ after very extensive training (10,000 training trials). Each entry $T_{jk} = |\langle D_k|C_j\rangle|^2$ of the transition matrix contains the probability of transiting

from a column input basis state $|C_j\rangle$ to a row output basis state $|D_k\rangle$. The target transition matrix is generated by taking the squared magnitudes of the cells in $U = \mathrm{e}^{-iH}$ with H defined by Eq. (11.4), and the estimated transition matrix is generated by taking the squared magnitudes of the cells in $\hat{U} = \mathrm{e}^{-iH_t}$ where H_t is the Hamiltonian learned from Eq. (11.3) after $t = 10,000$ trials. The target transition matrix is displayed below on the left as the matrix T and the estimate is displayed on the right as the matrix $\hat{T}$:

$$T = \begin{bmatrix} 0.4825 & 0.0631 & 0.1307 & 0.3238 \\ 0.0631 & 0.4825 & 0.3238 & 0.1307 \\ 0.1307 & 0.3238 & 0.2874 & 0.2581 \\ 0.3238 & 0.1307 & 0.2581 & 0.2874 \end{bmatrix},$$

$$\hat{T} = \begin{bmatrix} 0.4845 & 0.0633 & 0.1290 & 0.3232 \\ 0.0633 & 0.4823 & 0.3252 & 0.1292 \\ 0.1290 & 0.3252 & 0.2880 & 0.2579 \\ 0.3232 & 0.1292 & 0.2579 & 0.2898 \end{bmatrix}.$$

Comparing the two transition matrices, it can be seen that the estimate is very close to the target after extensive training. In fact, it was already reasonably close after only 1000 trials.

In sum, Eq. (11.3) provides a learning model to estimate a unitary matrix that transforms probability amplitudes expressed in one basis into probability amplitudes expressed in another basis. This learning model is new and it has not been empirically tested yet. However, it could be tested by training human learners with pairs of input and output probability distributions, and finally testing the human learners by presenting only the input distribution and asking the learner to generate the output distribution.

How can a person learn which unitary transform to use for different circumstances or situations? This is an issue that was raised earlier in Chapter 10. The next learning algorithm describes how to learn to apply control U gates to learn complex condition–action sequences.

11.3 Quantum reinforcement learning

For the third type of quantum learning, consider updating the amplitudes assigned to control U gate actions based on rewards and punishments, as done with reinforcement learning algorithms. A quantum reinforcement learning model for accomplishing this task was first proposed by Dong *et al.* (2008). The algorithm is based on the quantum information search algorithm originally proposed by Grover (1997). The quantum reinforcement learning algorithm does not require a quantum computer – instead, it can be directly used to learn to perform practical sequential decision-making tasks. The basic ideas are summarized below. First we review the basic concepts of the Markov decision-learning paradigm used in machine learning (Sutton & Barto, 1998) and then we describe the quantum reinforcement learning model.

11.3.1 Markov decision process paradigm

As described in Chapter 10, a POMDP consists of four sets: the first is a set $E = \{e_1, \ldots, e_n\}$ of environmental states, the second is a set $A = \{a_1, \ldots, a_m\}$ of mutually exclusive and exhaustive actions, the third is a set O of observations, and the fourth is a set R of rewards. In this chapter, for simplicity, we restrict the discussion to a Markov decision process (MDP) in which the states are directly observable, so that we do not distinguish between the sets E and O and dispense with the set O. The MDP also includes the two classic probability functions: a transition probability function $P_E : E \times A \times E \to [0, 1]$ that takes an environmental state and an action at one step in time and probabilistically selects a new state for the next step in time; and a reward function $P_R : E \times A \times R \to [0, 1]$ that takes a state and action and probabilistically delivers a reward or punishment. The agent applies a policy, which specifies the appropriate action to take for each state. The goal of a reinforcement learning algorithm is to learn a policy that maximizes the future discounted expected rewards.

In particular, the popular Q-learning algorithm works as follows (see Sutton and Barto (1998)). Define the estimate of the expected discounted future rewards for each state and action at time t as $Q(e_j, a_k, t)$. Suppose the last action taken at time t changed the state to e and the immediate reward $r(t)$ was obtained. Then the new estimate for each state is updated according to

$$Q(e_j, a_k, t+1) = (1-\eta) \cdot Q(e_j, a_k, t) + \eta \cdot \left[r(t) + \gamma \cdot \max_l Q(e, a_l, t) \right],$$

where η is a learning rate parameter and γ is a discount rate parameter. The term in square brackets is the Q-learning "reward" signal. Then the next action is probabilistically selected based on its expected future reward value (see Sutton and Barto (1998)). However, standard reinforcement learning algorithms suffer some problems, including slow learning in complex environments and sensitivity to parameters that balance exploration versus exploitation of the environment in the action selection rule. Therefore, new ideas for improving performance are still greatly needed in this area.

11.3.2 Quantum action selection

The basic idea is that the current environmental state puts the agent in a superposition state over the set of possible actions. The superposition state is a vector in an m-dimensional space spanned by m orthonormal basis vectors denoted $|a_k\rangle$, $k = 1, \ldots m$, and each basis vector corresponds to one of the actions. If the current state is e_j, then the superposition state over actions is

$$|\psi_j\rangle = \sum_{k=1}^{m} \psi_{jk} \cdot |a_k\rangle,$$

with two constraints on the amplitudes: $\psi_{jk} = 0$ for any action that is not available from state e_j and, given the previous constraint, we also require $|\psi_j\rangle$ to

remain unit length. Then the probability of taking action a_k from state e_j equals $\left|\psi_{jk}\right|^2$. The key new idea is the learning rule for amplifying the amplitudes ψ_{jk} that experience high rewards. To describe this algorithm we will ignore the basis states assigned zero amplitudes, because their actions are impossible from a given state, and consider only the basis states that correspond to possible actions. Hereafter, the $m \times 1$ matrix ψ will refer to the amplitudes for m actions and each action is assumed to be a potential choice.

11.3.3 Amplitude amplification

The amplitude amplification algorithm is an extension by Brassard and Hoyer of Grover's (1997) search algorithm (Hoyer, 2000). The algorithm begins with any arbitrary initial amplitude distribution represented by the $m \times 1$ matrix ψ_0, but it is common to start with $\psi_{jk} = \frac{1}{\sqrt{m}}$ for m actions from state e_j. Define ψ_t as the $m \times 1$ matrix of amplitudes after experiencing t trials of training. Suppose action a_j was chosen on the last trial t. The amplitude for action a_j is amplified or attenuated in proportion to reward $[r(t) + \gamma \cdot \max_l Q(e, a_l, t)]$ experienced by taking that action. This is done as follows.

Define A_k as an $m \times 1$ matrix with zeros in every row except the row k corresponding to action a_k, which is set equal to one. This is essentially the coordinates corresponding to the basis vector $|a_k\rangle$. Next define the following two matrices:

$$Q_1 = I - \left(1 - \mathrm{e}^{i\phi_1}\right) \cdot \left(A_k \cdot A_k^{\dagger}\right),$$

$$Q_2 = \left(1 - \mathrm{e}^{i\phi_2}\right) \cdot \left(\psi_t \cdot \psi_t^{\dagger}\right) - I,$$

where ϕ_1, ϕ_2 are two learning parameters that control the amount of amplification or attenuation. The matrix Q_1 flips the sign of the target action and the matrix Q_2 inverts all the amplitudes around the average amplitude, and together these to act to amplify the target while having no effect (except normalization) on the non-targets. Then the new amplitude distribution is formed by

$$\psi_{t+1} = (Q_2 \cdot Q_1)^L \cdot \psi_t, \tag{11.5}$$

where the matrix power L indicates the integer number of applications of the update used on a single trial. The key idea of the learning algorithm is to relate the reward $[r(t) + \gamma \cdot \max_l Q(e, a_l, t)]$ to the parameters ϕ_1, ϕ_2, and L applied after each trial. One option is to fix $\phi_1 = \phi_2 = \pi \approx 3.1416$ and allow L to be the integer value of $c \cdot [r(t) + \gamma \cdot \max_l Q(e, a_l, t)]$, where $c > 0$ is a free parameter (Dong *et al.*, 2008). This essentially produces the original Grover updating algorithm. However, this restricts the algorithm to discrete jumps in amplitudes, and it is too restrictive for small numbers of actions. Another option is to set $L = 1$ and restrict $\phi_1 = \phi_2 = \phi \cdot \pi \approx \phi \cdot 3.1416$ and vary ϕ in proportion to $c \cdot [r(t) + \gamma \cdot \max_l Q(e, a_l, t)]$ within a range that gives monotonic change in amplitude. The latter method is illustrated in the following two examples.

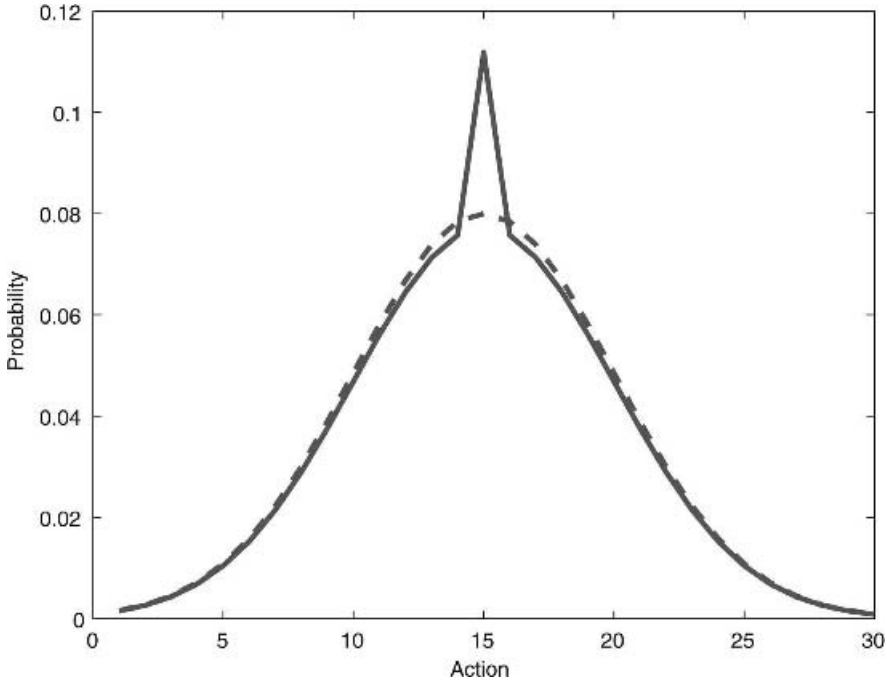

Figure 11.4 The smooth bell-shaped curve is the initial distribution and the curve with the spike is the distribution after amplification of action 15.

11.3.4 Examples of amplitude amplification

Two examples are provided below to illustrate how amplitude amplification works. Figure 11.4 depicts the amplification produced with $L = 1$ and $\phi = 0.15$ ($\phi_1 = \phi_2 = 0.15\pi$) for $m = 30$ actions and action 15 was updated. The initial distribution ψ_0 was approximately normally distributed, and the amplified distribution is shown with the spike at action 15. As can be seen, a single update modifies the selected action and leaves the remaining amplitudes the same except for renormalization. The computer code used to produce Figure 11.4 is presented in Appendix G.

Figure 11.5 shows the effect of varying $\phi_1 = \phi_2 = \phi \cdot \pi$ on the probability for the updated action. In this example, there are only $m = 3$ actions, and we set $L = 1$ and $\psi_k = \frac{1}{\sqrt{3}}$ initially so that the initial probability equals $\frac{1}{3}$. The top curve shows the amplification effect as a function of ϕ within the range $[1, 2]$ and note that $\phi = 1$ corresponds to $\phi_1 = \phi_2 = \pi$. The curve tends to oscillate and repeat itself if the range is extended. In this range, one could assume that ϕ is proportional to the reward signal.

Dong *et al.* (2010) extensively tested and compared the performance of the quantum reinforcement learning model with a standard reinforcement model using the popular "ϵ-greedy" algorithm (the best option is selected with probability

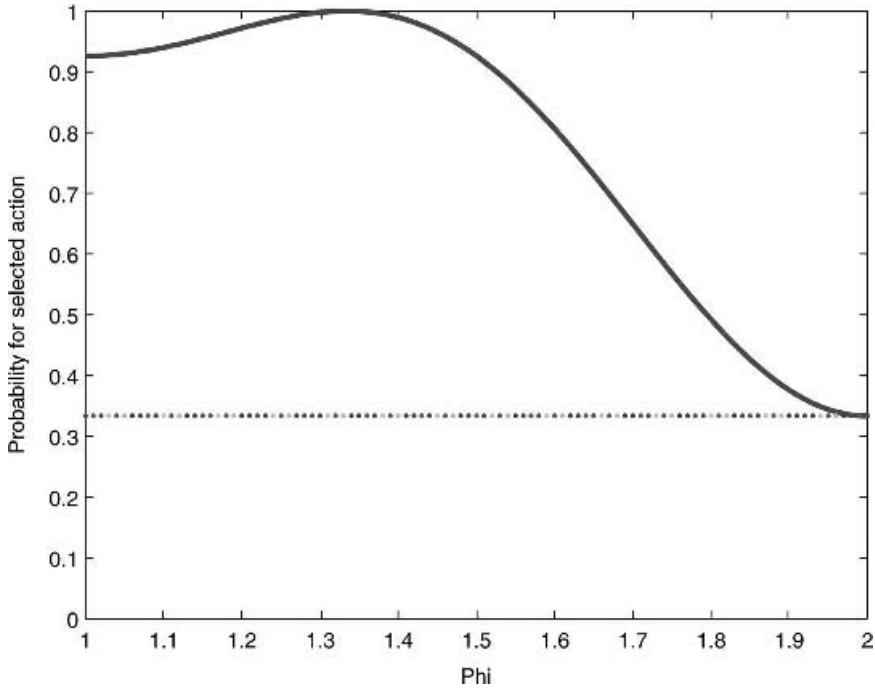

Figure 11.5 The top curve shows the probability assigned to the selected action after amplification; the bottom straight line is the probability before amplification. The difference is the amplification effect.

$1 - \epsilon$ and otherwise choice is random). The learning speeds for the two models were compared for different types of navigation environments (an artificial Markov world, a simulated robot navigation problem, and a real robot navigation problem) using a variety of learning rate parameters for the reinforcement learning model.[5] On the basis of these simulations and robotic experiments they reported that (a) the quantum reinforcement algorithm tended to reach convergence on the optimal solution faster than the traditional reinforcement learning model and, perhaps more importantly, (b) performance of the quantum reinforcement learning model was fairly robust with respect to variations in the learning rate parameter for reinforcement learning, whereas the performance of the traditional model was very sensitive to this learning parameter and worked well only within a restricted range. They concluded that the quantum action selection model improved performance by applying a more robust balance between exploration and exploitation. Amplitude amplification has also been used by Franco (2009b) to model human judgments.

[5]Dong *et al.* used a temporal difference learning model rather than a Q-learning model in their work. However, S.N. Balakrishan and K. Rajagopal (personal communication) replicated the original simulations by Dong and found that the Q-learning model works better.

11.4 Summary of learning models

This chapter addresses a fairly new topic on quantum learning models. Three different types of learning models were presented. One was Bayesian updating of quantum belief states (amplitudes) based on experience with independent and identically distributed observations on the values of variables. The second was to learn a unitary map between an input probability distribution (that describes the beliefs according to one set of basis vectors) into an output probability distribution (that describes the belief states by another set of basis vectors). In this case, learning was based on a gradient descent error function that measured the discrepancy between the predicted and the observed output probability distributions. The third type was a quantum reinforcement learning model that used an amplitude amplification algorithm (analogous to the Grover search algorithm) to update the probability amplitudes for actions in a Markov decision environment. Given the early stage of this research, the proposed models are designed to inspire future research rather than to be accepted as well-supported empirical models. Much more experimental and theoretical research needs to be done to extend these models, test their validity, and compare them with more traditional learning models.