Journal of Mathematical Psychology 53 (2009) 126–138

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

Theoretical tools for understanding and aiding dynamic decision making

Jerome R. Busemeyer [a,*], Timothy J. Pleskac [b]

[a] Department of Psychological and Brain Sciences, Indiana University, United States
[b] Department of Psychology, Michigan State University, United States

ARTICLE INFO

Article history:
Received 13 June 2007
Received in revised form
11 December 2008
Available online 6 February 2009

Keywords:
Dynamic decision making
POMDP
Separability
Sunk cost

ABSTRACT

Dynamic decisions arise in many applications including military, medical, management, sports, and emergency situations. During the past 50 years, a variety of general and powerful tools have emerged for understanding, analyzing, and aiding humans faced with these decisions. These tools include expected and multi-attribute utility analyses, game theory, Bayesian inference and Bayes nets, decision trees and influence diagrams, stochastic optimal control theory, partially observable Markov decision processes, neural networks and reinforcement learning models, Markov logics, and rule-based cognitive architectures. What are all of these tools, how are they related, when are they most useful, and do these tools match the way humans make decisions? We address all of these questions within a broad overview that is written for an interdisciplinary audience. Each description of a tool introduces the principles upon which it is based, and also reviews empirical research designed to test whether humans actually use these principles to make decisions. We conclude with suggestions for future directions in research.

Decision making is not getting any easier. Today's decisions are becoming more complex, with greater uncertainty, increasing time pressure, more rapidly changing conditions, and higher stakes. These dynamic types of decisions arise in many areas including military, medical, management, sports, and emergency situations, just to name a few (Bar-Eli & Raab, 2006; Klein, 1998). Examples of dynamic decisions within each of these areas include sequential information sampling, optional stopping of search, detection of change, navigational control, robotic tasks, health management, inventory control, portfolio management, emergency resource allocation, and many others.

A large box of tools has emerged for understanding, analyzing, and aiding humans faced with these decisions. This tool box includes expected and multi-attribute utility analyses (Keeney & Raiffa, 1993; Luce, 2000), game theory (Fudenberg & Tirole, 1991; Myerson, 1991), Bayesian inference (DeGroot, 1970) and Bayes nets (Pearl, 1988), decision trees and influence diagrams (Clemens, 1996), stochastic optimal control theory (Stengel, 1986), partially observable Markov decision processes (POMDP; Puterman (1994)), neural network (Haykin, 1999) and reinforcement learning models (Sutton & Barto, 1998), and rule-based cognitive architectures (Newell, 1990).

What are all of these tools, how are they related, when are they most useful, and do these tools match the way humans make decisions? Part of the answer to this question is that some of these tools (Markov decision problems, decision trees, influence diagrams, and Bayes nets) are models of the decision situation while the remaining are tools (expected and multi-attribute utility analyses, game theory, Bayesian inference, stochastic optimal control theory, POMDPS, reinforcement learning, rule based cognitive architectures) that can be used to analyze these situations to come to a (sometimes optimal) decision. There is more to these questions than this simple answer and in this article we aim to give a historical perspective which attempts to describe how these tools evolved across time. This perspective perhaps allows one to look ahead for new tools on the horizon or at least for directions where new tools need to be developed. The review also begins with simpler tools and works toward more complex ideas. Due to the scope of the review we will give a broad overview that briefly reviews many topics, rather than a detailed and thorough review of a specific topic.

1. A generic framework for dynamic decisions

First, let us provide a theoretical framework for discussing dynamic decisions. The framework is illustrated in Fig. 1 and can approximate a number of decisions with more specialized formal structures (Bertsekas, 1976, 1987). This system has a set of possible actions **A**, a set of system states **X**, an output set **Y**, and a set of uncertainty factors **E**, which in the figure is further decomposed into uncertainty about the state and uncertainty about the output.

To be concrete, consider a fisherman who at several different time points during the day (or more generally throughout the

* Corresponding address: Department of Psychological and Brain Sciences, Indiana University, 1101 E. Tenth Street, 47405 Bloomington, IN, United States.
E-mail address: jbusemey@indiana.edu (J.R. Busemeyer).

0022-2496/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmp.2008.12.007

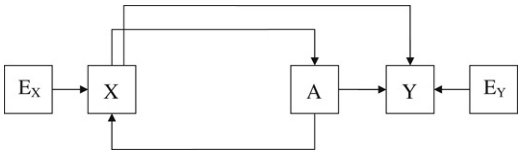

Fig. 1. A generic dynamic decision problem.

season) must decide where he is going to fish. [1] To do so, the fisherman selects among a set of alternatives or actions A (e.g., shallows in the Southern section of the lake or deep bay in the North). The decision at each time point reflects the abundance of fish at each location or the state of the system X and the fisherman's uncertainty in this abundance. It is also a function of the amount of fish caught or the output set Y, about which the fisherman may also be uncertain. For the time being, assume for simplicity, that each of the sets is finite (possibly large enough to closely approximate continua if needed) and that $x \in X, y \in Y, a \in A$, and $e \in E$. Additionally, suppose actions are taken and outputs are experienced at a finite but possibly large number of discrete time points $t \in \{t_1, \ldots, t_N\}$, where t_0 represents the current time point and t_N the furthest time point in the future used to make plans (which defines the decision horizon).

In this dynamic system the state of the system changes from time point to time point. For example, presumably our fisherman catches some fish and changes the state of the supply at some location. We characterize this change in the system with a system updating function $S : \mathbf{X} \times \mathbf{A} \times \mathbf{E} \rightarrow \mathbf{X}$ such that $x_{t+h} = S(x_t, a_t, e_t)$. That is, the state of the system at time point $t+h$ is a function of the previous state x_t, the action taken a_t, and the uncertainty at time point e_t, t. There is also an output function $M : \mathbf{X} \times \mathbf{A} \times \mathbf{E} \rightarrow \mathbf{Y}$ such that $y_t = M(x_t, a_t, e_t)$. Note that we assume that the state vector x contains all the information and memory about the past history needed to update the system for the future, and the output vector y contains both measurements of the states as well as payoffs for the actions.

The decision maker needs to form a policy, P, for choosing action a at time point t, which depends on the estimate of the current state at each time point, $a_t = P_t(x_t)$. The action in turn produces an output y_t via the output function at each time point. The resulting sequence of outputs $(y_1, y_2, \ldots, y_N)$ is assigned some path payoff value denoted $R(y_1, \ldots, y_N)$. In the fisherman example, the payoff might be the net value added in terms of the number of fish caught over the time and effort spent. Normally, the decision maker's objective is assumed to be to determine a policy P for selecting actions that maximizes the expected path payoff (averaged over paths for a given policy). That is the decision maker finds the P that maximizes $EU(P) = E[R(y_1, \ldots, y_N)|P]$ where the maximum is taken over feasible selections of possible P's.

The dynamic decision framework in Fig. 1 encompasses a broad range of decision situations including bandit problems (Berry & Fristedt, 1985). These problems or very similar problems have been popular in psychology (see Barron & Erev, 2003; Denrell, 2005, 2007; Erev & Barron, 2005; Steyvers, Lee, & Wagenmakers, 2009; Yechiam & Busemeyer, 2008). In these problems you are faced repeatedly with a choice among n different options, or actions. The options are different stochastic processes so that each choice produces a numerical reward determined by the respective stochastic process. If we have $n = 2$ options (play or not) then the problem can be analogous to its namesake: a slot machine. With multiple options the problems also reflect the decision our fisherman face about which location to fish in. The goal of the decision makers in these problems is to maximize their earnings over the entire (often finite) horizon. Note that a more general form of the bandit problem is a restless bandit problem where the payoff distributions are allowed to change over time (see Biele, Erev, & Ert, 2009; Whittle, 1988)

The challenge decision makers face with bandit problems is that the characteristics of each option are often unknown. However, decision makers are assumed to form an expectation from each option. Thus, there are two benefits from selecting an option: (1) immediate payoffs, and (2) information that can be used to develop one's expectation to make better choices in the future. Selecting the best option at any time point is often called the greedy action because it is exploiting the known values of the options. Selecting the other lesser valued options is an exploration choice because this allows decision makers to better form their expectation about the other options, which in the long run could be a more profitable selection if you happen to hold a mistaken belief about some of the options. This conflict between exploration and exploitation occurs in many real world decisions and is at the heart of the question decision-makers face in determining their strategy in these problems: how much to explore and how and when to exploit their knowledge. One way to answer this question is with dynamic programming described next (see Berry & Fristedt, 1985, for more details).

1.1. Solving for optimal policies

Finding an optimal policy for a dynamic decision can be a very complex problem, but the task becomes easier if one can adopt what is called a separable utility function. A utility function satisfies separability if the path payoff $R(y_1, \ldots, y_N)$ can be decomposed into the sum of the utilities of the separate contributions (assuming joint independence, see Koopmans (1960), postulate 3′). This function is typically formalized in terms of the sum of all possible future rewards, $R(y_1, \ldots, y_N) = \sum_t \gamma_t \cdot u(y_t)$. The parameter γ_t represents a weight for discounting payoffs depending on how far in the future they will occur. It represents the psychological principle that decision makers often prefer earlier rewards to later ones. In dynamic decision problems this discounting function is also important for convergence in infinite horizon problems where $N \rightarrow \infty$. In economics, the discount weight γ_t is usually set equal to an exponential function, $\gamma_t = \gamma^t$ $(0 < \gamma < 1)$. Psychological research, however, suggests that other discount functions are more descriptive of human preference (Loewenstein & Prelec, 1992).

Assuming separability (a topic we will return to in Section 4), the problem for the decision maker is to find a policy that maximizes the objective

$$EU(P) = E[R(y_1, \ldots, y_N)|P] = E\left[\sum_t \gamma_t \cdot u(y_t)|P\right] = \sum_t \gamma_t \cdot E[u(y_t)|P] \quad (1)$$

for $t \in \{t_1, t_2, \ldots, t_N\}$. The expectation is based on the probability that the system updates to a particular state. Dynamic programming methods solving for optimal policies with backward induction use this representation of utility and generally operates in the following manner.

Suppose the decision maker is facing the final decision to be made at time t_N. Expanding the summation on the right hand side of Eq. (1) we can isolate the last contribution to the expected path

[1] We thank Ido Erev for suggesting we use this fishing example inspired by Lane (1989).

payoff as follows:

$$E[R(y_1,\ldots,y_N)|P] = (\gamma_1 \cdot E[u(y_1)|P] + \cdots + \gamma_{N-1} \cdot E[u(y_{N-1})|P]) + \gamma_N \cdot E[u(y_N)|P]. \quad (2)$$

The expected payoff for the last stage, $E[u(y_N)|P]$, only depends on the action taken at this stage produced by policy P_N, $E[u(y_N)|P] = E[u(y_N)|P_N]$. To optimize this expected payoff, an action is selected, $a_N^* = P_N^*(x_N)$, that maximizes $E[u(M(x_N, P_N(x_N), e_N))]$ for each possible final state x_N. The value of this solution is defined as $V_N(x_N) = \gamma_N \cdot E[u(M(x_N, P_N^*(x_N), e_N))]$. At this point, we do not know what the final state will be, but we know it depends on the previous state and action, and so the maximum expectation over the final state can be described as

$$\gamma_N \cdot E[u(y_N)|P_N^*] = \sum p(S(x_{N-1}, a_{N-1}, e_{N-1}) = x_N) \cdot V_N(x_N),$$

where $p(S(x_{N-1}, a_{N-1}, e_{N-1}) = x_N)$ is the probability the system updates to x_N.

Having made this plan for the last stage, we can update Eq. (2) as follows

$$E[R(y_1,\ldots,y_N)|P] = (\gamma_1 \cdot E[u(y_1)|P] + \cdots + \gamma_{N-2} \cdot E[u(y_{N-2})|P]) + \gamma_{N-1} \cdot E[u(y_{N-1})|P_{N-1}] + \gamma_N \cdot E[u(y_N)|P_N^*], \quad (3)$$

and recede backward a step to identifying a maximizing policy to use in the second-to-last stage. However, because $E[u(y_N)|P_N^*]$ depends on x_{N-1} and a_{N-1} the optimal policy in the second-to-last stage $P_{N-1}^*(x_{N-1})$ must maximize $E[u(M(x_{N-1}, P_{N-1}(x_{N-1}), e_{N-1}))] + \gamma_N \cdot E[u(y_N)|P_N^*]$ for each state x_{N-1}. Define the value of this solution as

$$\begin{aligned} V_{N-1}(x_{N-1}) &= \gamma_{N-1} \cdot E[u(M(x_{N-1}, P_{N-1}^*(x_{N-1}), e_{N-1}))] + \gamma_N \cdot E[u(y_N)|P_N^*] \\ &= \gamma_{N-1} \cdot E[u(M(x_{N-1}, P_{N-1}^*(x_{N-1}), e_{N-1}))] + \sum p(S(x_{N-1}, P_{N-1}^*(x_{N-1}), e_{N-1}) = x_N) \cdot V_N(x_N). \end{aligned}$$

The maximum expectation over the final two states is then equal to

$$\gamma_{N-1} \cdot E[u(y_{N-1})|P_{N-1}^*] + \gamma_N \cdot E[u(y_N)|P_N^*] = \sum p(S(x_{N-2}, a_{N-2}, e_{N-2}) = x_{N-1}) \cdot V_{N-1}(x_{N-1}).$$

Having made this plan for the last two stages, we can update Eq. (3),

$$E[R(y_1,\ldots,y_N)|P] = (\gamma_1 \cdot E[u(y_1)|P] + \cdots + \gamma_{N-3} \cdot E[u(y_{N-3})|P]) + \gamma_{N-2} \cdot E[u(y_{N-2})|P_{N-2}] + \gamma_{N-1} \cdot E[u(y_{N-1})|P_{N-1}^*] + \gamma_N \cdot E[u(y_N)|P_N^*],$$

and back up to the third-to-last stage. The maximum expected payoff for the last three stages is obtained by selecting $P_{N-2}^*(x_{N-2})$ that optimizes $E[u(M(x_{N-2}, P_{N-2}(x_{N-2}), e_{N-2}))] + \gamma_{N-1} \cdot E[u(y_{N-1})|P_{N-1}^*] + \gamma_N \cdot E[u(y_N)|P_N^*]$ for each state x_{N-2}. We can define this solution as

$$V_{N-2}(x_{N-2}) = \gamma_{N-2} \cdot E[u(M(x_{N-2}, P_{N-2}^*(x_{N-2}), e_{N-2}))] + \sum p(S(x_{N-2}, P_{N-2}^*(x_{N-2}), e_{N-2}) = x_{N-1}) \cdot V_{N-1}(x_{N-1}).$$

Suppose the decision only has three stages so that $N = 2$ and we started in state $x_{N-2} = x_0$. Then the solution process is completed because all of the variables would be known at this point.

The whole process can be extended backwards step by step beyond three stages. Given that we have already found $\{P_N^*, P_{N-1}^*, \ldots, P_{N-k-1}^*\}$, and we know $V_{N-k-1}(x_{N-k-1})$, we can then back up another step by solving for $P_{N-k}^*(x_{N-k})$ to produce

$$V_{N-k}(x_{N-k}) = \gamma_{N-k} \cdot E[u(M(x_{N-k}, P_{N-k}^*(x_{N-k}), e_{N-k}))] + \sum p(S(x_{N-k}, a_{N-k}, e_{N-k}) = x_{N-k-1}) \cdot V_{N-k-1}(x_{N-k-1}).$$

The process we just described is the recursive form of the dynamic programming algorithm and it forms the basis for finding the optimal solution for many dynamic decisions. As can be seen from above, the dynamic programming algorithm breaks the value of stage k into the sum of the immediate expected utility and the expected utility for the future state of the system. This process depends on a separable utility function, which is quite a strong restriction, but necessary for using the recursive backward induction schemes to solve complex problems with long time horizons.

2. Utility theory

Utility theory is designed for evaluating and selecting courses of actions and is thus a tool that can be used to help find an optimal solution in dynamic decision problems. This includes evaluations involving risk or uncertainty of the outcomes, as well as evaluations involving multiple conflicting attributes or objectives. For example, for our fisherman, utility theory would be used for evaluating whether or not the expected number of fish caught outweighs the cost of resources spent to catch the fish.

Utility theory has a long history stretching all the way back to the 1700's with some inspirational work by Daniel Bernoulli (see Bernoulli (1954/1738)). A rigorous modern theory of utility was first formulated by von Neumann and Morgenstern (1947). This theory was designed to provide a representation of preferences over risky actions, where a risky action is defined by a probability distribution over a set of possible uncertain outcomes. For example, which of two investments would you prefer: a risky investment a_R which has a .50 probability of returning a gain equal \$100,000 or a .50 probability of producing loss equal to \$50,000; versus a safe investment a_S which returns a gain of \$25,000 for sure?

To address this investment question, von Neumann and Morgenstern proposed a small set of intellectually appealing axioms, which, if accepted, leads to a formula for meaningfully assigning real numbers or utility values to gambles via their expected utility, $EU = \sum p_i \cdot u(y_i)$. Where p_i is the probability of outcome y_i and $u(y_i)$ is the utility of outcome y_i, and the sum is across all possible outcomes. For example, one axiom is transitivity: If you prefer action a_1 to a_2, and you prefer action a_2 over a_3, then you should prefer action a_1 over a_3. Transitivity seems quite reasonable and one might question the rationality of a person who violates this axiom (but see Rieskamp, Busemeyer, and Mellers (2006)). Once transitivity and the other von Neumann and Morgenstern axioms are met, then a utility function can be constructed that can then be used to compare your preferences over risky actions (Luce, 2000). This utility function assures that choices made based on the assigned utilities will satisfy the axioms. Thus, utility theory defines what most decision theorists accept as the theory of rational or normative decision making.

Referring to our question of investment preference, if you used a utility function such as $u(y) = \sqrt{y}$ for $y \geq 0$, and $u(y) = -\sqrt{|y|}$ for $y < 0$ then you could rationally prefer investment a_S, with $EU(a_S) = \sqrt{\$25{,}000} = 158$, to investment a_R with $EU(a_R) = .5\sqrt{100{,}000} - .5\sqrt{50{,}000} = 46$, even though the expected dollar value for a_R is much larger. This is an example of what a decision theorist would call a 'risk averse' utility function. Risk aversion is not unreasonable — many of us buy insurance, which is also a 'risk averse' decision.

2.1. Extensions of utility theory

Three important extensions followed the seminal work by von Neumann and Morgenstern. First, the original theory was limited to 'actions under risk' with probabilities provided by some valid source or derived from some mechanism in an objective or uncontroversial manner. For example, suppose the return depends

on the flip of a fair coin, then it is uncontroversial to assign equal probabilities. Rarely does a decision maker face a situation with such crisp and precisely known probabilities. More often one is faced with 'actions under uncertainty' in which you have to rely on your personal beliefs about uncertain events. For example, suppose the return depends on whether a democrat or a republican wins the next election. Nobody really knows the exact probability for this event, and people may differ in their beliefs about the likelihoods. Savage (1954) extended the axioms of expected utility theory to allow for personal probabilities for uncertain events.

Second, the original theory was limited to gambles with outcomes that can be described by some uni-dimensional value like money. However, many decisions involve multiple and possibly conflicting dimensions such as quality and cost of a consumer product, or quality and cost of medical care, location and size of an apartment, or speed versus accuracy of a decision. Keeney and Raiffa (1993) developed an axiomatic foundation for a comprehensive utility theory for multi attribute outcomes.

The third extension is more recent and fundamental because it changes the axioms upon which the original rational theory was founded. One of the axioms assumed by all of the above utility theories is the independence axiom, which can be loosely stated as follows: If one prefers action a_1 to a_2, then one should also prefer a probabilistic mixture of action a_1 with a_3 to the same probabilistic mixture of action a_2 with a_3. According to classic utility theory, the common component of the mixture, a_3, should cancel out and not affect your preference. However, thirty years of experimental tests of this independence axiom indicate that people generally do not obey it (Allais, 1979; Kahneman & Tversky, 1979). Apparently the independence axiom is too restrictive, and so utility theorists have weakened the axiom and produced a new formula for rank ordering preferences called rank dependent utility theory (see Luce, 2000, for a review).

3. Bayesian inference and bayes nets

Bayesian inference is used to infer the probability of a hypothesis or the probability of a state of the world, based on previously collected evidence or data. For example, this tool would be useful for our fisherman who needs to infer the distribution of fish in areas of the lake based on environmental cues or information from other people. Other applications of Bayesian inference might be to infer the hidden intent of an enemy based on previous behaviors that you have observed or infer a disease state of an individual based on laboratory tests.

Expected utility theory and Bayesian inference form the two pillars upon which decision theory is built, and they integrate in a rational and elegant manner to provide prescriptions for decision making. This integration should not be taken for granted because it is difficult to rationally justify methods for integrating beliefs with values to inform decisions using alternative models of reasoning under uncertainty. The latter includes the Neymann–Pearson hypothesis testing procedure, Zadeh's fuzzy set theory (Zimmerman, 2001), or Dempster–Shaffer belief systems (Shafer & Pearl, 1990).

To see the close link between Bayesian inference and Expected Utility, consider for example, the deferred decision problem (also called the sequential sampling problem). The deferred decision problem can be viewed as a dynamic version of signal detection theory (Smith, 2000), and it is highly relevant to problems such as pattern recognition and target detection. Suppose a system can be in either a signal state (x_S) or a noise state (x_N) with prior probabilities of $p(x_S)$ and $p(x_N)$. You have available one of two possible actions: a_h = act hostile, a_f = act friendly. Your payoff depends on both the true state and the action: $u(a_h, x_S)$ is the gain for a hit, $u(a_h, x_N)$ is the loss for a false alarm, $u(a_f, x_S)$ is the loss for a miss, and $u(a_f, x_N)$ is the gain for a correct rejection. Your decision is informed by collecting a series of independent and identically distributed samples, $[e_1, e_2, \ldots, e_t]$, whose distribution depends on the state, and each sample is assumed to cost a fixed amount c.

Using the notation from our generic dynamic system described above, the previous state of the system x_{t-1} is updated after sampling each new observation e_t to a new state x_t, using the posterior odds form of Bayes rule:

$$x_t = \frac{p(x_S|[e_1, \ldots, e_t])}{p(x_N|[e_1, \ldots, e_t])} = x_{t-1} \cdot \frac{p(e_t|x_S)}{p(e_t|x_N)},$$

$$\text{with } x_0 = \frac{p(x_S)}{p(x_N)}.$$

The posterior probabilities of each hypothesis are

$$p(x_S|[e_1, e_2, \ldots, e_t]) = \frac{x_t}{1 + x_t} \quad \text{and}$$

$$p(x_N|[e_1, e_2, \ldots, e_t]) = \frac{1}{1 + x_t}.$$

At each time point you are faced with a choice among three options: stop and act hostile, stop and act friendly, or sample another observation. The expected utilities for each of three actions at time t are:

$$\text{EU (stop and choose } a_h|x_t) = \tfrac{x_t}{1+x_t} u(a_h, x_S) + \tfrac{1}{1+x_t} u(a_h, x_N)$$

$$\text{EU (stop and choose } a_f|x_t) = \tfrac{x_t}{1+x_t} u(a_f, x_S) + \tfrac{1}{1+x_t} u(a_f, x_N)$$

$$\text{EU (continue to sample } |x_t) = -c + \textstyle\sum p(e_{t+1}) \cdot V(x_{t+1})$$

where $V(x_{t+1})$ is the expected utility of following the optimal policy from next evidence state x_{t+1} after observing a new observation e_{t+1}. The optimal decision is to choose the option that is the maximum of these three expected utilities at each point in time. Using the dynamic programming methods discussed earlier, it can be shown that the optimal strategy is to continue sampling until the posterior probability exceeds a fixed threshold (DeGroot, 1970; Edwards, 1965; Rapoport & Burkheimer, 1971). This idea forms the rational foundation for random walk and diffusion models of decision making, which are popular in cognitive science (Busemeyer & Townsend, 1993; Laming, 1968; Link & Heath, 1975; Ratcliff, 1978) as well as neuroscience (Bogacz, Brown, Moehlis, Holmes, & Cohen, 2006; Schall, 2003; Shadlen & Newsome, 2001).

3.1. Bayes nets

An important extension of Bayesian inference is a graphical tool called Bayes nets (Pearl, 1988). They are particularly useful in representing complex uncertain situations to find the joint distribution of the underlying events. Specifically, a Bayes net is an augmented directed acyclic graph represented by a set of vertices and a set of directed edges joining vertices. Each vertex contains a probability distribution that depends on the parent edges (an example is given below). Thus, Bayes nets become efficient at representing complex uncertain situations when the parents of the vertices are relatively small in number so that the joint distribution can be reconstructed from the product of conditionally independent components.

Fig. 2 provides a simple example of a Bayes net that contains 5 vertices. For example, e_1 could represent a disease state, which affects the probabilities assigned to the values of measurements of medical symptoms e_2 and e_3, and these measurements influence the likelihood of the type of a diagnosis at e_4, and finally this diagnosis affects the probability of different treatments at e_5.

The random variable e_1 does not depend on any other variable in this system so that we only need to define $p(e_1 = x_i)$ for all i. The random variable e_2 depends on e_1, so that we need to define $p(e_2 = x_i|e_1 = x_j)$ for all i, j. Similarly, for e_3 we need to define $p(e_3 = x_i|e_1 = x_j)$ for all i, j. The random variable e_4 depends on both e_2 and e_3 but it is conditionally independent of e_1 so we need to define $p(e_4 = x_i|e_2 = x_j, e_3 = x_k)$ for i, j, k. Finally e_5 only depends on e_4 so we only need to define $p(e_5 = x_i|e_4 = x_j)$ for all i, j. All the joint probabilities can then be computed from the

product of conditionals

$$p(e_1 = x_i, e_2 = x_j, e_3 = x_k, e_4 = x_l, e_5 = x_m)$$
$$= p(e_1 = x_i) \cdot p(e_2 = x_j | e_1 = x_i) \cdot p(e_3 = x_k | e_1 = x_i)$$
$$\times p(e_4 = x_l | e_2 = x_j, e_3 = x_k) \cdot p(e_5 = x_m | e_4 = x_l).$$

In sum, if each node had 5 values, then we need to define $(5-1) + 5(5-1) + 5(5-1) + 25(5-1) + 5(5-1) = 164$ probabilities to reconstruct the entire set of $(5^5 - 1) = 3124$ joint probabilities. All sorts of other conditional probabilities, such as $p(e_5 = x_i | e_1 = x_j)$ or $p(e_1 = x_i | e_4 = x_j)$, can be inferred from the joint distribution.

3.2. Hierarchical Bayes nets

Of course, 164 probability estimates is still a large number to determine, and so how can these estimates be determined? Furthermore, how can one determine the structure of a Bayes net? Bayesian learning schemes are being developed to solve these problems (Neapolitan, 2004). One interesting example is the use of hierarchical Bayes nets (Tenenbaum, Griffiths, & Kemp, 2006). At the top of the hierarchy are probabilities assigned to hypotheses about the form of various structures for a network (e.g., assign a probability to a tree form). The next layer assigns probabilities to specific structures given a hypothesized form (e.g. assign a probability to a specific type of tree structure, given that it is a tree form). The final layer assigns probabilities to the branches of a specific tree structure. Using this hierarchical structure, data can be used to inform not only the edges of a particular Bayes net, but also the most appropriate abstract form of the Bayes net. This generalization of Bayes nets provides robustness with respect to the representation of knowledge. Bayes nets are not limited to discrete variables, and a very elegant theory can be formulated based on multivariate normal distributions.

3.3. Human inference

There is quite a large empirical literature on probability inference and causal inference (see Kahneman, 1982, for a review of probability inference and Novick & Cheng, 2004, for a review of causal reasoning). This research addresses the question of whether or not people actually reason according to the rules of probability. To get an idea of these findings, it is worthwhile to point out at least two important results from each area. A large number of experiments on causal inference have been conducted using simple 2×2 tables of frequencies of the form: putative cause present or absent, effect observed or not. What is frequently found in this paradigm is that people over weight the frequencies in the 'cause present–effect present' cell. If this frequency is high, people tend to infer a contingent statistical relationship when in fact none is present; and when this frequency is low, they tend to underestimate a true contingency. Many consider this a sign of irrationality in human inference; however others have tried to explain this from a Bayesian point of view (Anderson, 1990).

Perhaps the most famous finding related to probabilistic inference is the conjunctive fallacy (Tversky & Kahneman, 1983). For example, suppose you are told that Harry is a 56 year old overweight business man who is under high pressure and who has a lot of stress in his life. Given this fact, you are asked to judge the likelihood of three future scenarios for Harry: (a) Harry will die within the next year; (b) Harry will have a heart attack within the next year; (c) Harry will have a heart attack and die within the next year. Not surprisingly, most people judge (b) to be most likely, but surprisingly they judge (c) to be more likely than (a). This judgment is a violation of a basic property of probability because p (die within next year) $= p$ (heart attack and die within next year) $+ p$ (no heart attack and die within next year) $\geq p$ (heart attack and die within next year). Once again this has been interpreted as another dramatic sign of irrationality in human inference, however again there is a possible Bayesian explanation for this result (Tenenbaum & Griffiths, 2001).

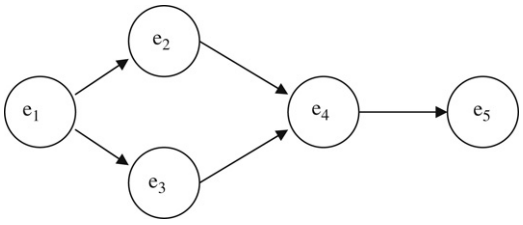

Fig. 2. A Bayes net with 5 vertices.

4. Decision trees and path utility

The complexity in dynamic decision problems makes them difficult to conceptualize for both lay and expert decision makers. A simple but important tool that decision theorists use to represent relatively short time horizon dynamic decisions is a decision tree (Clemens, 1996; von Winterfeldt & Edwards, 1986). This tool is limited to short time horizons because the branches in the tree grow in number very fast with extensions of the decision horizon. But, when applicable, decision trees are a useful tool for understanding and reducing the cognitive load of dynamic decision problems. In reviewing this tool we will also address two important theoretical issues that arise in dynamic decisions: one is the *separability* of the utility function, and the second is the reliance on *backward induction*. Similarly, two interesting psychological issues arise in this context: one is called the *sunk cost* problem, and the second is the problem of *dynamic inconsistency*.

Consider the example shown in Fig. 3, which represents a dynamic decision that a fisherman may face at a fishing contest. Suppose you bet another person that you can catch more weight than the other. The decision process begins at node a_1 where you can decide to travel to a lake in the Westside of town (West or left at a_1) or in the Eastside (East or right at a_1). Imagine that you estimate that your opponent will catch 2 or 3 lbs. of fish with equal probability.

Following any of these outcomes you potentially face a decision about how you want to proceed. For the purpose of this illustration, consider what you might do if the opponent catches a 2 pound fish ($y_1 = -2$, left at e_1) after you go East. The state of the tournament is now you are 2 pounds behind'. At this point, you have to decide if you are going to be aggressive and go for a 3 lb. fish (left branch at a_2) which you might catch with probability .15 or not. Now consider instead the alternative where you go for a 2 pound fish (right branch). To make the scenario interesting, imagine if you catch a 2 pound fish you expect to have time to go after another fish. Therefore, if you catch a 2 pound fish (right at e_3) then you may catch another fish or not. And this fish could be 0, 1, 2, or 3 pounds with probabilities .4, .25, .25, and .10, respectively.

4.1. Separable utility functions and sunk cost

To decide what actions to take in this problem, you need to select a payoff function for each path, where a path is a sequence of actions and events starting at the top node and ending at a bottom node. Suppose you use the difference in total pounds between what you and your opponent have accumulated as the utility function for each path. That is, the utility of an outcome equals the net weight accumulated, and the path payoff equals the net weight produced by a path. Note that this utility function satisfies the separability assumption: $R(y_1, y_2) = u(y_1) + u(y_2)$. Using this utility function, we can *ignore* the 'sunk cost' of our opponent catching a two pound fish when we make our decision to travel to a new spot at node a_2 and we simply focus on the future consequences. To make a decision at a_2 we compare the expected utilities associated with each future path following a_2:

$EU(\text{ left at } a_2) = (.15)(3) = .45 < EU(\text{right at } a_2) = (.2)(.4)(2) + (.2)(.25)(3) + (.2)(.25)(4) + (.2)(.1)(5) = .61$. The

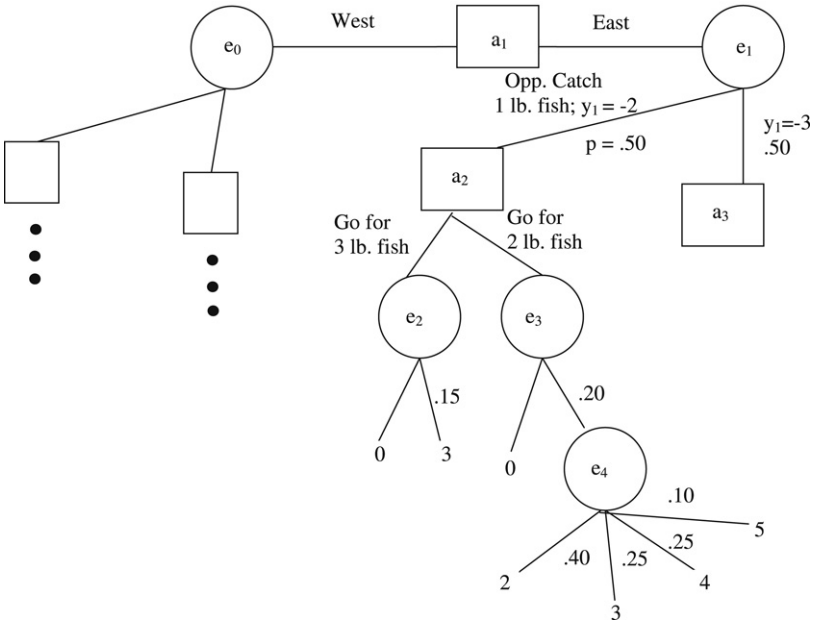

Fig. 3. A decision tree involving a sequence of actions a fisherman in a competition to catch the most weight in fish might face.

difference in expected utilities, .61 − .45 = .16, favors going right at a_2. This action is valid in this case because the 'sunk cost' of losing 2 pounds to your competitor is a common consequence that would simply cancel out when we compare the utilities of going left and right at decision node a_2 *after* incorporating the common loss of two pounds: $EU(\text{left at } a_2) = -2 + (.15)(3) = -1.55 < EU(\text{right at } a_2) = -2 + (.2)(.4)(2) + (.2)(.25)(3) + (.2)(.25)(4) + (.2)(.1)(5) = -1.39$, and the difference is $-1.39 - (-1.55) = .16$, the same as before.

Suppose instead your goal is to win the fishing contest (catch the most weight) rather than maximize the net weight caught. That is if you are down 2 pounds you would see catching a 2 pound fish as equivalent to catching no fish at all because you would not win the contest. Formally, $R(y_1, y_2, \ldots, y_N) = 1$ if $y_1 + y_2 + \cdots + y_N > 0$, and zero otherwise. If the utility function is separable then,

$$R(y_1 = 0, y_2 = 2) > R(y_1 = 0, y_2 = 1) \rightarrow$$
$$u(y_1 = 0) + u(y_2 = 2) > u(y_1 = 0) + u(y_2 = 1) \rightarrow$$
$$u(y_2 = 2) > u(y_2 = 1) \rightarrow$$
$$u(y_1 = -2) + u(y_2 = 2) > u(y_1 = -2) + u(y_2 = 1) \rightarrow$$
$$R(y_1 = -2, y_2 = 2) > R(y_1 = -2, y_2 = 1).$$

But, in fact this new utility function assigns $R(y_1 = -2, y_2 = 2) = R(y_1 = -2, y_2 = 1) = 0$. Therefore, this utility function violates the additive rule required for separability.

Consider the implications of this violation in the fishing contest decision tree in Fig. 3. At node a_2, with this different goal of simply winning the contest, if we now mistakenly ignore the 'sunk cost' of our opponent catching a fish and evaluate only the future consequences we find that $EU(\text{left at } a_2) = (.15)R(3) = (.15)(1) = .15 < EU(\text{right at } a_2) = (.2)(.4)R(2) + (.2)(.25)R(3) + (.2)(.25)(4) + (.2)(.1)(5) = (.2)(.4)(1) + (.2)(.25)(1) + (.2)(.25)(1) + (.2)(.1)(1) = .20$ and we would choose to go for a 2 pound fish. But, if we correctly include the −2 loss from the pounds fish our opponent caught at node e_1 into our net weight, then the correct utilities are $EU(\text{left at } a_2) = (.15)R(3-2) = (.15)(1) = .15 > EU(\text{right at } a_2) = (.2)(.4)R(2-2) + (.2)(.25)R(3-2) + (.2)(.25)(4-2) + (.2)(.1)(5-2) = (.2)(.4)(0) + (.2)(.25)(1) + (.2)(.25)(1) + (.2)(.10)(1) = .12$ and we would choose to go for the 3 pound fish (left at a_2).

Several psychological experiments have been conducted to examine whether or not decision makers ignore past outcomes and base their decisions solely on future consequences (Arkes & Blummer, 1985). Frequently it is found that people are not willing to ignore these 'sunk costs' and this is usually considered irrational behavior (see Dawes & Hastie, 2001). However, as this example shows, whether or not one should consider past payoffs when making future decisions depends on whether or not the utility function is separable.

4.2. Backward induction and dynamic consistency

In order to decide what to do at decision node a_1 and more generally find an optimal solution to problems represented in decision trees like our fishing example, decision theorists use what is called a backward induction algorithm. You plan ahead, and work backward from the possible future end nodes to the imminent decision. Even though you are currently trying to decide what to do at node a_1, you need to first plan ahead and decide what action to take at node a_2. Using net pounds caught as our utility function we find that EU (left at a_2) $< EU$(right at a_2), so that going right at a_2 is optimal, and thus $EU(a_2) = EU(\text{right at } a_2)$. When we back up and finally evaluate action a_1, we compute $EU(\text{right at } a_1) = (.5) \cdot EU(a_2) + (.5) \cdot EU(a_3)$. Finally, to determine the final optimal policy, you would also need to evaluate the action of going left at a_1 (going to the lake on the Westside). As discussed earlier in Section 1, this recursive computation of expected utility lies at the heart of dynamic programming algorithms.

To be effective, the backward induction algorithm relies on the assumption that the decision maker will be dynamically consistent, i.e., planned actions are faithfully carried out. For example, suppose your selection at node a_1 depends on your plan to go right at node a_2. In that case, if and when you arrive at node a_2, you need to actually go right as planned. If you change your mind at this later stage, you would be dynamically inconsistent, destroying the plan. Thus, dynamic consistency is a key assumption underlying the use of backward induction, and backward induction is the basic principle underlying the use of dynamic programming.

Several recent experiments have been designed using simple gambles to empirically determine whether or not people actually satisfy dynamic consistency (see Cubitt, Starmer, & Sugden,

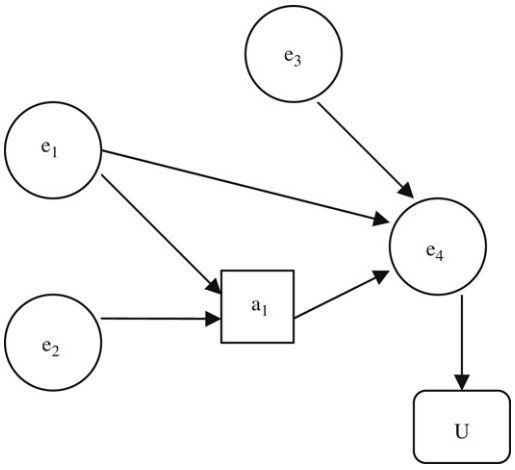

Fig. 4. An influence diagram with decision and event node dependencies.

2004, for a review). These empirical results show that dynamic consistency frequently fails even when simple gambles are used (involving real money) and participants are given full information about the probabilities and payoffs before making their plans. Furthermore, this dynamic inconsistency far exceeds the rate of inconsistency for repeated choices, and so the result cannot be explained by choice inconsistency alone. Apparently the decision makers' utility function changes as they progress down a decision tree.

4.3. Influence diagrams

On the one hand, Bayes nets provide a convenient graphical representation for making inferences but not for evaluating decisions. On the other hand, decision trees are useful for planning decisions, but they do not provide a cogent representation of the dependencies among events for making inferences. These two tools can be combined into a common framework for representing decision problems which call for making inferences and decisions using what are called influence diagrams. Essentially, decision nodes and the dependencies entailed by these decisions are combined with event nodes and dependencies among events to form an integrated representation (Clemens, 1996; Howard & Matheson, 2005). Fig. 4 shows a situation where the decision at node a_1 depends on knowing events e_1 and e_2, and the event at e_4 depends on events e_1, e_3, and a_1. The final utility depends on event e_4. The calculation of utility for each action produced by the influence diagram turns out to be the same as that for the decision tree, and so the same optimal policy will be selected. The gain from using influence diagrams is mainly obtained by more efficient representation and the improvement in understanding.

5. Game theory and equilibrium solutions

Most of the situations and tools we have talked about until now have assumed that the outcomes a decision maker experienced were not directly influenced by the decisions opposing decision makers made. Game theory is a tool that can be used when this is not true. That is when the outcomes in these situations for one decision maker are also a function of the choices others make (Fudenberg & Tirole, 1991; Myerson, 1991). Games involve intelligent adversaries (players) who also have preferences over the outcomes of the situation you are in and have partial control of the outcomes. Decision trees, such as those in Fig. 3, can easily be augmented with decision nodes that distinguish the various players for both simultaneous and sequential games forming what is called extensive form representations of the game (Luce & Raiffa, 1957).

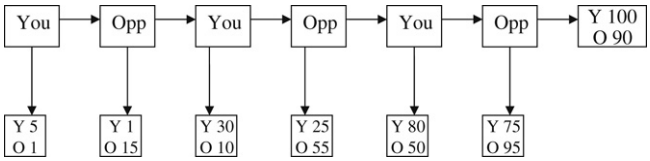

Fig. 5. Two player, six stage centipede Centipede Game.

In game theory, players are typically assumed to be fully rational and only wish to maximize their self interest (Luce & Raiffa, 1957). The rational strategy under these assumptions is a Nash equilibrium strategy (Nash, 1951): If all the players adopt the equilibrium strategy, then no player has any incentive to unilaterally change strategies. For sequential games, the Nash equilibrium is often found by solving the game in extensive form using the subgame perfect equilibrium method (Harsanyi & Selten, 1988). This method forms a set of strategies that constitute a Nash equilibrium in every subgame of the original game. The latter is found by backward induction by means of finding the optimal strategy for the last play of the game, then find the optimal strategy for the second to last play, given the optimal strategy of the last play, and continue until this process works back to the beginning.

Fig. 5 illustrates a game first developed by Rosenthal (1981) often studied in the laboratory called the centipede game (see Camerer (2003)). In this case, you are playing against your opponent and you have the first move. If you go down, you win $5 and the opponent gets only $1 and the game is over; but if you go right, then your opponent makes the next move. If your opponent goes down, you win $1 and you opponent wins $15 and the game is over; but if your opponent goes right, then you make the next move.

Note that if you and your opponent progress to the final stage, then you could earn up to $100 while your opponent could earn up to $90. However, because the game is dominance solvable, working backwards, we see that reaching the last node is not what should happen. Starting at the last possible decision stage, if the game ever reaches this point then your opponent has the final move, and she is better off going down (earning $95) rather than moving right (earning $90). Assuming that this is what your opponent will do at the last stage, if you reach the second to last stage, you will be better off going down (earning $80) rather than going right to the last stage (earning $75). Working backwards then finally leads to the conclusion that at the first move you will go down (earning $5) rather than moving right (earning $1). Thus, as game theory assumes, if players are rational and have full knowledge the prediction for the game is grim and perhaps paradoxical: it will end on the first move with you earning only $5 and your opponent earning only $1, even though you both could cooperate and potentially earn very much more.

Empirically, during centipede games with low stakes (possible earnings between $0 and $25.60) people quite frequently move to the right with the tendency to go down increasing as one approaches the end of the tree (McKelvey & Palfrey, 1992). However, when the stakes are increased 100-fold (possible earnings range between $0 and $2560.00), the number of players are increased from 2 to 3, and players are rematched over iterated games, behavior approaches the equilibrium play of stopping on the first node (Parco, Rapoport, & Stein, 2002). An adaptive learning model that assumes an updating of individual choice probabilities accounts for these results over and above other static and dynamic models (Rapoport, Stein, Parco, & Nicholas, 2003)

5.1. Backward induction revisited

We see in this example, just as with the analysis of decision trees, that backward induction is an important assumption underlying the solutions for equilibrium strategies. In other areas, like bargaining with another agent, experimental economists have

found evidence that people do not even implement an information search strategy consistent with backward induction. Instead they tend to search for information in a forward looking manner (see Johnson, Camerer, Sen, & Rymon, 2002).

6. Stochastic optimal control theory

Control theory is most useful for planning paths that are evolving smoothly across time and space. In this case, the set of states, actions, and uncertainties are continuous rather than finite and discrete. The time index is also usually treated as a continuum, although often this is approximated by a discrete but fine grained set of equally spaced time points. A common example for which this tool is useful is navigating a vehicle in space and time, and so this theory has been particularly useful for uninhabited vehicle control (Astrom & Murray, 2008). But, control theory is not limited to this situation, and it can be applied much more broadly to include other examples such as managing a patient's health across time, or managing fishing resources over time.

Modern control theory started developing rapidly in the 1950's, and it developed somewhat independently of decision theory. Initially this independent development prevented researchers from seeing the close connections between these two theories. Modern control theory is now based more directly on the same principles as decision theory: the goal is to solve for a sequence of controls that minimize some expected cost function over time.

Consider for example the very popular linear stochastic optimal control problem. For this application, we need to extend our generic dynamic system to allow for continuous rather than discrete states, actions, and uncertainties; but for simplicity, we will continue to work with a fine grain set of discrete time points. The system updating function S is given by the matrix equation

$$x(t+h) = S(x_t, a_t, e_t) = F \cdot x(t) + G \cdot a(t) + e_x(t),$$

where $e_x(t)$ is a noise term. At this point let us assume that the state is directly observable so that $y_t = (x_t, a_t)'$. The objective is to select the control inputs at each time to minimize the expected deviations around a target or desired trajectory denoted d_t:

$$E\left[\sum_t (x_t - d_t)' Q_x (x_t - d_t) + a_t' Q_y a_t\right].$$

The matrices Q_x and Q_y are used to differentially weight deviations on each coordinate, and differentially weight deviations from target versus control cost.

This is equivalent to defining the goal as an expected utility maximization problem, where the utility that results from each action is defined as

$$u(y_t) = -[(x_t - d_t)' Q_x (x_t - d_t) + a_t' Q_y a_t].$$

Decision theorists would call this an ideal point utility model (Coombs & Avrunin, 1977). Note that this objective function assumes a separable utility function and dynamic programming methods (using backward induction) are used to find the optimal solution. The optimal solution is a linear function, say $C \cdot x$, of the state variable (Stengel, 1986), so that the system equation becomes

$$\begin{aligned} x(t+h) &= F \cdot x(t) + G \cdot C \cdot x(t) + e_x(t) \\ &= (F + G \cdot C) \cdot x(t) + e_x(t) = H \cdot x(t) + e_x(t). \end{aligned}$$

6.1. Bayesian connection with state space models

A close connection between stochastic optimal control theory and Bayesian inference emerges when the state is hidden and not directly observable. General state space models allow for the possibility that the state is updated by the linear system

$$x(t+h) = S(x_t, a_t, e_t) = H \cdot x(t) + e_x(t).$$

However, the output is limited to a fallible set of measures:

$$y(t) = M(x_t, a_t, e_t) = M \cdot x(t) + e_y(t).$$

The noise terms, $e_x(t)$ and $e_y(t)$, are assumed to be normally distributed and uncorrelated. In this case we need to estimate the hidden or latent state from the limited and fallible measures. Assume that $e_x(t) \sim N(0, \Sigma_x)$ and $e_y(t) \sim N(0, \Sigma_y)$. Then the Kalman filter provides a Bayesian rule for state updating (Meinhold & Sinpurwalla, 1983). Suppose that the posterior distribution at time $t - h$ is $f(x_{t-h}|y_{t-h}) = N(\mu_{t-h}, \Sigma_{t-h})$. Then the prior distribution for time t is given by

$$f(x_t) = N(H \cdot \mu_{t-h}, R_t), \quad \text{with } R_t = H \cdot \Sigma_{t-h} \cdot H' + \Sigma_x.$$

Finally, the posterior distribution at time t, after incorporating the new observation y_t is

$$\begin{aligned} f(x_t|y_t) &= N(\mu_t = H\mu_{t-h} + R_t M'(\Sigma_y + MR_t M')^{-1}(y_t - MH\mu_{t-h}), \\ \Sigma_t &= R_t - R_t M'(\Sigma_y + MR_t M')^{-1} MR_t). \end{aligned}$$

The whole process starts with some initial guesses for (μ_0, Σ_0), and once these are specified, the distribution of future states evolves according to the above recursive formula.

6.2. Human control over dynamic systems

There have been a number of psychological studies to examine human performance on dynamic control problems (see Busemeyer, 2002; Sterman, 1994, for reviews). Sterman (1989) found that when subjects tried to manage a simulated production task, they produced costs 10 times greater than optimal, and their decisions induced costly cycles even though the consumer demand was constant. Brehmer and Allard (1991) found that when subjects tried to allocate resources to manage a simulated forest fire, they frequently allowed their headquarters to burn down despite desperate efforts to put the fire out. Kleinmuntz and Thomas (1987) found that when subjects tried to manage their simulated patients' health, they often let their patients die while wasting time waiting for the results of non-diagnostic tests and performed more poorly than a random benchmark. In these experiments, the participants are provided all the information that they need to solve the optimal policy, and so why do people have such difficulty controlling these dynamic situations?

6.3. Learning to control

One of the obvious reasons that it is so difficult for people to control dynamic systems is a cognitive one. That is, even when all the necessary information for determining the optimal policy is provided, people do not have the cognitive resources and knowledge to explicitly solve this complex problem. Instead, they need to implicitly learn how to control dynamical systems through extensive hands on experience with feedback. Past studies reveal that overall performance improves with extensive training (Gonzalez, Lerch, & Lebiere, 2003; Rapoport, 1966), and subjective policies tend to evolve over trial blocks toward the optimal policy (Jagacinski & Miller, 1978; Jagacinski & Hah, 1988). Thus, humans are bounded rational decision makers (Simon, 1982), whose performance is limited by their information processing and learning capacities.

Three different frameworks for modeling human learning processes in dynamic decision tasks have been proposed. One approach is based on production rule models such as ACT-R (Anderson, 1990; Anderson et al., 2004), SOAR (Laird, Newell, & Rosenbloom, 1987), EPIC (Meyer & Kieras, 1997), and CLARION (Sun, Zhang, & Mathews, 2006). These models assume that a large set of condition–action rules is incrementally learned from experience to control a system. For example, Anzai (1984) used a

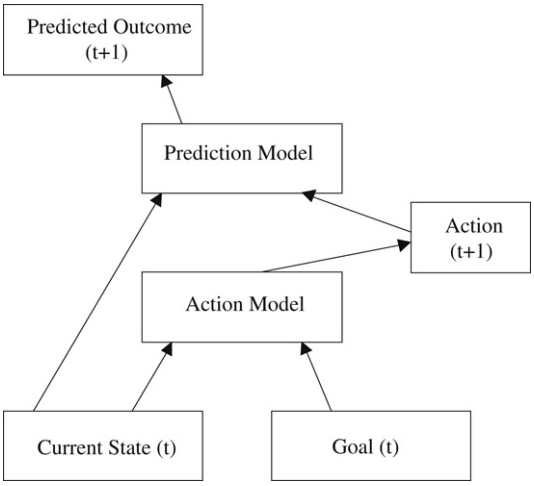

Fig. 6. A neural network for learning to control a dynamic system.

production rule system to describe how humans learn to navigate a simulated ship.

A second approach is based on instance or exemplar or case based learning. These models assume that whenever an action leads to a successful outcome, then the preceding situation and the successful response are stored together in memory. On any given trial, stored instances are retrieved on the basis of similarity to the current situation, and the associated response is applied to the current situation. Exemplar models were employed by Dienes and Fahey (1995) to describe how humans learn to control a simulated sugar production task, and by Gonzalez et al. (2003) to describe how humans learn to control a hydraulic plant. Gilboa and Schmeidler (1995) have used the model in economic applications for consumer choice.

A third approach is based on the use of supervised artificial neural network models (Haykin, 1999). Neural networks were originally developed by psychologists to model concept learning, pattern recognition, and language learning (Grossberg, 1988; Rumelhart & McClelland, 1986). Supervised neural networks require the environment to serve as a teacher by presenting corrective feedback that indicates how the outcome produced deviates from some target or goal state for the system. Back propagation learning algorithms use this feedback to adjust the weights that perform the mappings from inputs to predictions to improve the network's accuracy.

Fig. 6 illustrates one neural network model that uses two types of input nodes: one representing the current state of the environment and the other representing the current goal for the task. These inputs feed into the next layer of hidden nodes that compute the next action given the current state and goal. The action and the current state then feed into another layer of hidden nodes, which is used to predict the consequence of the action given the current state. The connections from the current state and action to the prediction hidden layer are learned by back propagating prediction errors; and the connections from the current state and current goal to the action hidden layer are learned by back propagating deviations between the observed outcome and the goal state. A neural network model of this form was developed by Gibson, Fichman, and Plaut (1997) to describe learning in a sugar production task, and it provided good accounts of participants' performance during both training and subsequent generalization tests under novel conditions. Neural networks are now commonly used in engineering to automatically learn to control dynamical systems based on large training data sets (see Bertsekas & Tsitsiklis, 1996; Jordan & Rumelhart, 1992; Miller, Sutton, & Werbos, 1991).

7. Markov decision processes

Markov decision processes (MDP's) are most useful for planning complex paths that evolve in a very discontinuous and discrete manner across long time horizons. This nicely complements control theory, which requires a smooth evolution across time and space. In this case, the set of states, actions, uncertainties are all finite and discrete. The time index is also usually treated as discrete. Examples for which this tool has been used include target search and identification, weapons allocation, robotic tasks, medical diagnosis, and marketing (see Cassandra, 1998 for a review).

Markov decision processes were developed by operations researchers a little bit later than the previously described tools (Howard, 1960), with a close link to decision theory, but somewhat independently of control theory. Although there are several very fundamental differences between MDP's and stochastic optimal control theory, these two theories also share a lot of important principles. As an important example of their similarity, note that the commonly used linear stochastic control system described earlier is also a Markov process (with continuous rather than discrete states).

The generic model for dynamic decision making described in Section 1 actually includes a MDP, although the latter are not usually described in this way. We chose this formulation to facilitate making connections between MDP's and stochastic optimal control theory. In the typical formulation of an MDP, what we call the system function $S(x_t, a_t, e_t)$ is represented by a state to state probability transition matrix $T(x_t, a_t, x_{t+1})$. This transition matrix defines the probability of transiting to the next state x_{t+1} given the previous state x_t, and action a_t. The two are related by setting $p(S(x_t, a_t, e_t) = x_{t+1}|x_t, a_t) = T(x_t, a_t, x_{t+1})$. Also the reward function $r(x_t, a_t)$ in the typical MDP is related to the output function $M(x_t, a_t, e_t)$ of our generic dynamic decision model by the expectation $r(x_t, a_t) = E[u(y_t)|x_t, a_t]$. The optimal policies for MDP's are solved using the dynamic programming algorithm described in Section 1.

7.1. Bayesian connection with Partially Observable MDP's (POMDP)

An important extension of MDP's was achieved allowing for the possibility that the states of the system are hidden and not directly observable. For example, referring back to our fishing example in Section 1, a more plausible case is that fishermen do not know the actual state of the fish stock in each location. In terms of our generic dynamic system, the decision maker (fisherman) only observes y_t (amount of fish caught). Markov decision processes assume that x_t is included in y_t, so that the state is observable. Partially observable MDP's (POMDP's) assume that this is not the case, and instead x_t must be inferred from y_t using Bayes rule (see Littman, 2009, this issue for a review). This corresponds to the state space model in control theory, which uses the Kalman filter (also derived from Bayes rule) to estimate the state from the measurements.

POMPD models assume that there is a transition matrix, $O(x_t, a_t, o_t)$ to the observation o_t given the state and action. In terms of our generic dynamic system model, this is obtained by setting $O(x_t, a_t, o_t) = p(M(x_t, a_t, e_t) = o_t|x_t, a_t)$. Using Bayes rule, we can derive the updating rule for estimating the state as follows. Suppose our posterior probability for each state after incorporating y_{t-h} is $p(x_{t-h}|y_{t-h})$. Then the prior probability after action a_t but before incorporating y_t is given by

$$p(x_t|a_t) = \sum p(x_{t-h}|y_{t-h}) \cdot T(x_{t-h}, a_t, x_t).$$

The posterior probability after taking into consideration y_t is then equal to

$$p(x_t|a_t, o_t) = p(x_t|a_t) \cdot O(x_t, a_t, o_t)/p(o_t|a_t),$$