Reinforcement Learning

Chris Watkins

Department of Computer Science
Royal Holloway, University of London

July 27, 2015

Plan 1

- Why reinforcement learning? Where does this theory come from?
- Markov decision process (MDP)
- Calculation of optimal values and policies using dynamic programming
- Learning of value: TD learning in Markov Reward Processes
- Learning control: Q learning in MDPs
- Discussion: what has been achieved?
- Two fields:
  - Small state spaces: optimal exploration and bounds on regret
  - Large state spaces: engineering challenges in scaling up

What is an 'intelligent agent'?

What is intelligence?

Is there an abstract definition of intelligence?

Are we walking statisticians, building predictive statistical models of the world?

- If so, what types of prediction do we make?

Are we constantly trying to optimise utilities of our actions?

- If so, how do we measure utility internally?

Predictions

Having a world-simulator in the head is not intelligence. How to plan?

Many types of prediction are possible, and perhaps necessary for intelligence.

Predictions conditional on an agent committing to a goal may be particularly important.

In RL, predictions are of total future 'reward' only, conditional on following a particular behavioural policy.
No predictions about future states all!

A wish-list for intelligence

A (solitary) intelligent agent:

- Can generate goals that it seeks to achieve. These goals may be innately given, or developed from scattered clues about what is interesting or desirable.
- Learns to achieve goals by some rational process of investigation, involving trial and error.
- Develops an increasingly sophisticated repertoire of goals that it can both generate and achieve.
- Develops an increasingly sophisticated understanding of its environment, and of the effects of its actions.

+ understanding of intention, communication, cooperation, ...

Learning from rewards and punishments

It is traditional to train animals by rewards for 'good' behaviour and punishments for bad.

Learning to obtain rewards or to avoid punishment is known as 'operant conditioning' or 'instrumental learning'.

The behaviourist psychologist B.F. Skinner (1950s) suggested that an animal faced with a stimulus may 'emit' various responses; those emitted responses that are *reinforced* are strengthened and more likely to be emitted in future.

Elaborate behaviour could be learned as 'S-R chains' in which the response to each stimulus sets up a new stimulus, which causes the next response, and so on.

There was no computational or true quantitative theory.

Thorndike's Law of Effect

> *Of several experiments made in the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond.*

Thorndike, 1911.

Law of effect: a simpler version

...responses that produce a satisfying effect in a particular situation become more likely to occur again in that situation,

and responses that produce a discomforting effect become less likely to occur again in that situation

Criticism of the 'law of effect'

It is stated like a natural law – but is it even coherent or testable?

Circular : What is a 'satisfying effect'? How can we define if an effect is satisfying, other than by seeing if an animal seeks to repeat it? 'Satisfying' later replaced by 'reinforcing'.

Situation : What is 'a particular situation' ? Every situation is different!

Preparatory actions : What if the 'satisfying effect' needs a sequence of actions to achieve? e.g. a long search for a piece of food? If the actions during the search are unsatisfying, why does this not put the animal off searching?

Is it even true? : Plenty of examples of actions of people repeating actions that produce unsatisfying results!

Preparatory actions

To achieve something satisfying, a long sequence of unpleasant preparatory actions may be necessary.

This is a problem for old theories of associative learning:

- Exactly when is reinforcement given? The last action should be reinforced, but should the preparatory actions be inhibited, because they are unpleasant and not immediately reinforced?
- Is it possible to learn a long-term plan by short-term associative learning?

Solution: treat associative learning as adaptive control

Dynamic programming for computing optimal control policies was developed by Bellman (1957), Howard (1960), and others.

Control problem is to find a control policy that minimises average cost, (or maximises average payoff).

Modelling associative learning as adaptive control introduces a new psychological theory of associative learning that is more coherent and capable than before.

Finite Markov Decision Process

Finite set $\mathcal{S}$ of states; $|\mathcal{S}| = N$

Finite set $\mathcal{A}$ of actions; $|\mathcal{A}| = A$

On performing action a in state i:

- probability of transition to state j is P_{ij}^{a}, independent of previous history.
- on transition to state j, there is a (stochastic) immediate reward with mean $R(i, a, j)$ and finite variance.

The *return* is the discounted sum of immediate rewards, computed with a *discount factor* γ, where $0 \leq \gamma \leq 1$.

Transition probabilities

When action a is performed in state i, P_{ij}^a is the probability that the next state is j

These probabilities depend only on the current state and not on the previous history (Markov property).

For each a, P_{ij}^a is a Markov transition matrix; for all i, $\sum_j P_{ij}^a = 1$

To represent transition probabilities (aka dynamics) we may need up to $A(N^2 - N)$ parameters.

State - action - state - reward

An agent 'in' a MDP repeatedly:

1. *Observes* the current state s
2. *Chooses* an action a and performs it
3. Experiences/causes a transition to a new state s'
4. *Receives* an immediate reward r,which may depend on s, a, and s'

Agent's experience completely described as a sequence of tuples

$$\langle s_1, a_1, s_2, r_1 \rangle$$
$$\langle s_2, a_2, s_3, r_2 \rangle$$
$$\dots$$
$$\langle s_t, a_t, s_{t+1}, r_t \rangle$$
$$\dots$$

Defining immediate reward

Immediate reward r can be defined in several ways.

Experience consists of $\langle s, a, s', r \rangle$ tuples

r may depend on any subset of s, a, and s'

But s' depends only on s and a, and s' becomes known only after action is performed.
For *action choice*, only $\mathbb{E}[r \mid s, a]$ is relevant.

Define $R(s, a)$ as:

$$R(s, a) = \mathbb{E}[r \mid s, a] = \sum_{s'} P^a_{ss'} \mathbb{E}[r \mid s, a, s']$$

Reward and return

Return is a sum of rewards. Sum can be computed in three ways:

Finite horizon : there is a terminal state that is always reached, on any policy, after a (stochastic) time T:

$$v = r_0 + r_1 + \cdots + r_T$$

Infinite horizon, discounted rewards : for discount factor $\gamma < 1$,

$$v = r_0 + \gamma r_1 + \cdots + \gamma^t r_t + \cdots$$

Infinite horizon, average reward : Process never ends, but need assumption that MDP is irreducible for all policies:

$$v = \lim_{t \to \infty} \frac{1}{t}(r_0 + r_1 + \cdots + r_t)$$

Return as total reward: finite horizon problems

Termination must be guaranteed.

- Shortest-path problems.
- Success of a predator's hunt.
- Number of strokes to put the ball in the hole in golf.
- Total points in a limited duration video game.

If number of time-steps is large, then learning becomes hard since the effect of each action may be small in relation to total reward.

Return as total discounted reward

Introduce a *discount factor* γ, with $0 \leq \gamma < 1$.
Define return :

$$v = r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 r_3 + \cdots$$

We can define return from time t:

$$v_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$$

Note the recursion:

$$v_t = r_t + \gamma v_{t+1}$$

What is the meaning of γ?

Three interpretations:

- A 'soft' time horizon to make learning tractable.
- Total reward, with $1 - \gamma$ as probability of interruption at each step.
- Discount factor for future utility. γ quantifies how a reward in the future is less valuable than a reward now.

Note that γ may be a random variable and may depend on s, a, and s'.

e.g. where γ is interpreted as a discount factor for utility, and different actions take different amounts of time.

Policy

A policy is a rule for choosing an action in every state: a mapping from states to actions. A policy, therefore, defines behaviour.

Policies may be deterministic or stochastic: if stochastic, we consider policies where the random choice of action depends only on the current state s

Defined over whole state space.

'Closed-loop' behaviour: observe state, then choose action given observed state.

When following a policy, the policy makes the decisions: the sequence of states is a Markov chain. (In fact the sequence of $\langle s, a, s', r \rangle$ tuples is a Markov chain.)

Value function for a policy

The *value* of a state s given policy π is the expected return from starting in s and following π thereafter by taking action $\pi(s)$ in each state s visited.

Can think of π as a 'composite action'.

$$V^\pi(i) \quad = \mathbb{E}[r_0 + \gamma r_1 + \cdots \mid \text{start in } i \text{ and follow } \pi]$$

By linearity of expectation, we have the N linear equations:

$$V^\pi(i) = R(i, \pi) + \gamma \sum_j P^\pi_{ij} V^\pi(j)$$

so that V^π is given by:

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

Policy improvement lemma

Suppose that the value function for policy π is V^π, and for some state i, there is a non-policy action $b \neq \pi(i)$, such that

$$R(i,b) + \gamma \sum_j P_{ij}^b V^\pi(j) = V^\pi(i) + \epsilon, \text{ where } \epsilon > 0$$

Consider the policy π' such that

$$\pi'(i) = b \text{ and } \pi'(j) = \pi(j) \text{ for } j \neq i$$

Then, by the Markov property,

$$V^{\pi'}(i) \geq V^\pi(i) + \epsilon$$
$$V^{\pi'}(j) \geq V^\pi(j) \text{ for all } j \neq i$$

If you see a quick profit, or short-cut, according to your current value function, take it!

Policy improvement algorithm

Given: MDP and initial policy π

Repeat

Calculate value function:

$$V^{\pi} = (I - \gamma P^{\pi})^{-1} R^{\pi}$$

Improve policy: For all i,

$$\pi'(i) \leftarrow \arg\max_{a} \left(R(i,a) + \gamma \sum_{j} P_{ij}^{a} V^{\pi}(j) \right)$$

$\pi \leftarrow \pi'$

Until there has been no change in π

Bellman optimality equations

Policy iteration must terminate with V^*, π^* which cannot be further improved, and which satisfy:

$$V^*(i) = \max_a \; R(i,a) + \gamma \sum_j P_{ij}^a V^*(j)$$

$$\pi^*(i) = \arg\max_a \; R(i,a) + \gamma \sum_j P_{ij}^a V^*(j)$$

These conditions are necessary for V^* and π^* to be optimal, but are they sufficient?

Are V^* and π^* unique? Could there be 'locally optimal' policies?

These questions may be easily answered with a different solution method, value iteration.

Action values

Define

$$Q^{\pi}(i,a) = R(i,a) + \gamma \sum_{j} P_{ij}^{a} V^{\pi}(j)$$

From policy improvement lemma,

$$\max_{a} Q^{\pi}(i,a) \geq V^{\pi}(i)$$

and Bellman optimality equations become:

$$Q^{*}(i,a) = R(i,a) + \gamma \sum_{j} P_{ij}^{a} \max_{b} Q^{*}(j,b)$$

Note that Q^{*} represents both the optimal value function and policy

$$V^{*}(i) = \max_{a} Q^{*}(i,a) \quad \text{and} \quad \pi^{*}(i) = \arg\max_{a} Q^{*}(i,a)$$

Value iteration: a different optimisation procedure

Define a sequence of finite-horizon MDPs, working backwards in time, with Q tables $Q_0, Q_{-1}, Q_{-2}, \ldots$

$Q_0(i, a)$ is a table of arbitrary payoffs

$$Q_{-1}(i, a) = R(i, a) + \gamma \sum_j P_{ij}^a \max_b Q_0(j, b)$$

$$\vdots$$

$$Q_{-(t+1)}(i, a) = R(i, a) + \gamma \sum_j P_{ij}^a \max_b Q_{-t}(j, b)$$

Q_{-t} is optimal for the the t stage process from $-t$ to 0 that terminates with final payoffs $Q_0(i, a)$ (Proof by induction)

Max-norm contraction property

Consider two value-iteration sequences starting from Q_0 and Q'_0.

$$\max_{i,a}(Q_{-(t+1)}(i,a) - Q'_{-(t+1)}(i,a)) \leq \gamma \max_{i,a}(Q_{-t}(i,a) - Q'_{-t}(i,a))$$

As $t \to \infty$, $\|Q_{-t} - Q'_t\|_\infty \to 0$

Provided that the policy Markov chains are irreducible, Q_{-t} and Q'_{-t} must converge to the same limit Q^*, which satisfies the Bellman optimality equations.

In particular,

$$\|Q_{-(t+1)} - Q^*\|_\infty \leq \gamma \|Q_{-t} - Q^*\|_\infty$$

Summary of optimality properties

Uniqueness

Q^* and hence V^* are unique; π^* is unique up to actions with equal Q^*

Deterministic policy

optimal returns can be achieved with a deterministic policy (no need for stochastic action choice)

No local maxima

In a policy with sub-optimal values, there is always *some* state where policy improvement gives a better choice of action.

Free interleaving of updates

States need not be updated in a fixed systematic scan; as long as all states are updated sufficiently often, the following updates will converge:

$$V(i) \leftarrow R(i, \pi(i)) + \gamma \sum_j P_{ij}^{\pi(i)} V(j) \text{ and}$$

$$\pi(i) \leftarrow \max_a R(i, a) + \gamma \sum_j P_{ij}^a V(j)$$

or

$$Q(i, a) \leftarrow R(i, a) + \gamma \sum_j P_{ij}^a \max_b Q(j, b)$$

Modes of control

An *agent* in a MDP can choose its actions in several ways:

Explicit policy agent maintains a table π of policy actions (or action probabilities)

Q-greedy agent maintains a table of Q; in state i chooses $\arg\max_a Q(i, a)$, or some stochastic function of these Q values

One step look-ahead Agent maintains V, P, and R. In state i, chooses $\arg\max_a R(i, a) + \gamma \sum_j P_{ij}^a V(j)$

Sample based planning Agent maintains V, P, and R, and samples a forward search tree to estimate action-values.

Function approximation

What may happen if we use a Q-greedy policy π from $\tilde{Q}$ which approximates Q^* ?

Suppose $\|\tilde{Q} - Q^*\|_\infty \leq \epsilon$, then

$$V^\pi \geq V^* - \frac{2\epsilon}{1-\gamma}$$

Suppose function approximation is used *at each stage* in policy iteration, with max-norm error ϵ; then

$$V^\pi \geq V^* - \frac{2\epsilon}{(1-\gamma)^2}$$

These bounds are tight!
Usually we are not so unlucky.

Temporal difference (TD) learning

The problem: estimate value from experience.

An agent follows a policy in an unknown MDP, visits a sequence of states and receives rewards.

$s_1, r_1, s_2, r_2, \ldots, s_t, r_t, \ldots$

How can agent 'learn' or estimate the values of the states from this experience?

Idea 1: Model based learning. Keep statistics on state transition probabilities and rewards, estimate a model of the process, and then calculate the value function from the model.

Not very 'neural'!
Difficult to extend to function approximation methods.

Model-free estimation: backward-looking TD(1)

Idea 2: for each state visited, calculate the *return* for a long sequence of observations, and then update the estimated value of the state.

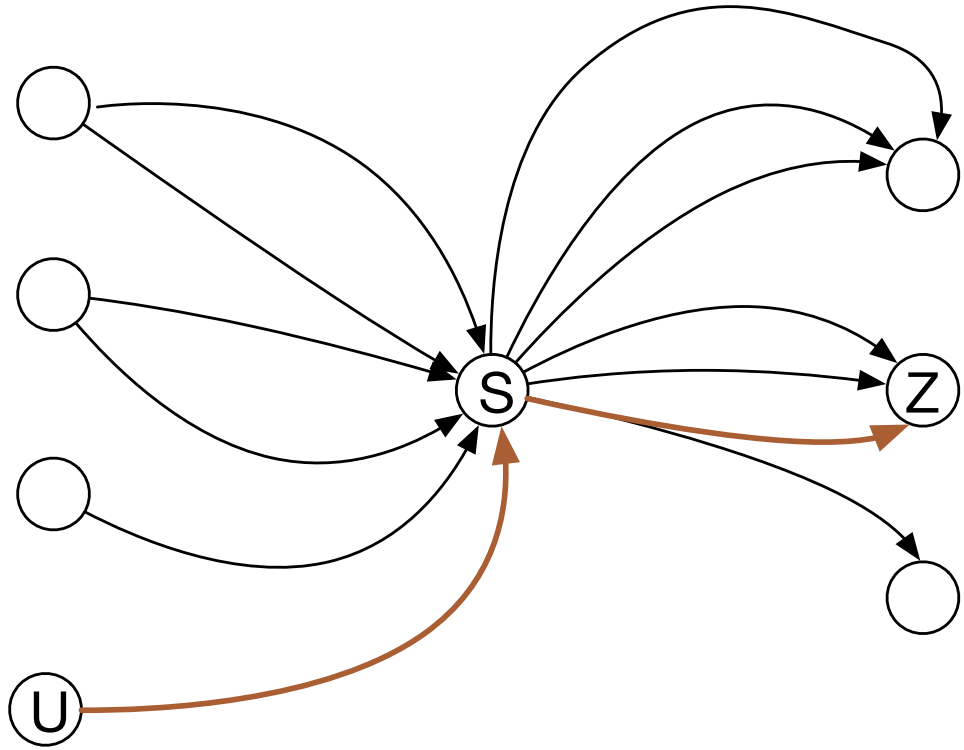
Set $T \gg \frac{1}{1-\gamma}$. For each state s_t visited, and for a learning rate α,

$$V(s_t) \leftarrow (1-\alpha)V(s_t) + \alpha(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^T r_{t+T})$$

Problems:

- Return estimate only computed after T steps; need to remember last T states visited. Update is late!
- What if process is frequently interrupted, so that only small segments of experience available?
- Estimate is unbiased, but could have high variance. Does not exploit Markov property!

TD(1) estimates may have high variance



The TD(1) value estimate for the rarely-visited state U is based on rewards along the single brown path, which visits S.

But V(S) can be well estimated from other experiences: this additional information is not used in the TD(1) estimate for U.

Model free estimation: short segments of experience

We can think of V as a Q table with only one action: the value-iteration update

$$V(i) \leftarrow R(i) + \gamma \sum_j P_{ij} V(j)$$

cannot increase $\|V - V^*\|_\infty$.

(It may make $V(i)$ less accurate, but cannot increase max error.)

But a *model-free* agent cannot perform this update because it does not know P; what about the stochastic update, for small $\alpha > 0$:

$$\begin{aligned} V(s_t) &\leftarrow (1-\alpha)V(s_t) + \alpha(r_t + \gamma V(s_{t+1})) \\ &= V(s_t) + \alpha(r_t + \gamma V(s_{t+1}) - V(s_t)) \\ &= V(s_t) + \alpha\delta_t \end{aligned}$$

TD(0) learning

Define the *temporal difference prediction error*

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

Agent maintains a V-table, and updates $V(s_t)$ at time $t+1$:

$$V(s_t) \leftarrow V(s_t) + \alpha\delta_t$$

Simple mechanism; solves problem of short segments of experience.

Dopamine neurons seem to compute δ_t !

Does TD(0) converge?
Can be proved using results from theory of stochastic approximation, but simpler to consider a visual proof.