Reinforcement Learning: introduction

Chris Watkins

Department of Computer Science
Royal Holloway, University of London

November 25, 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.

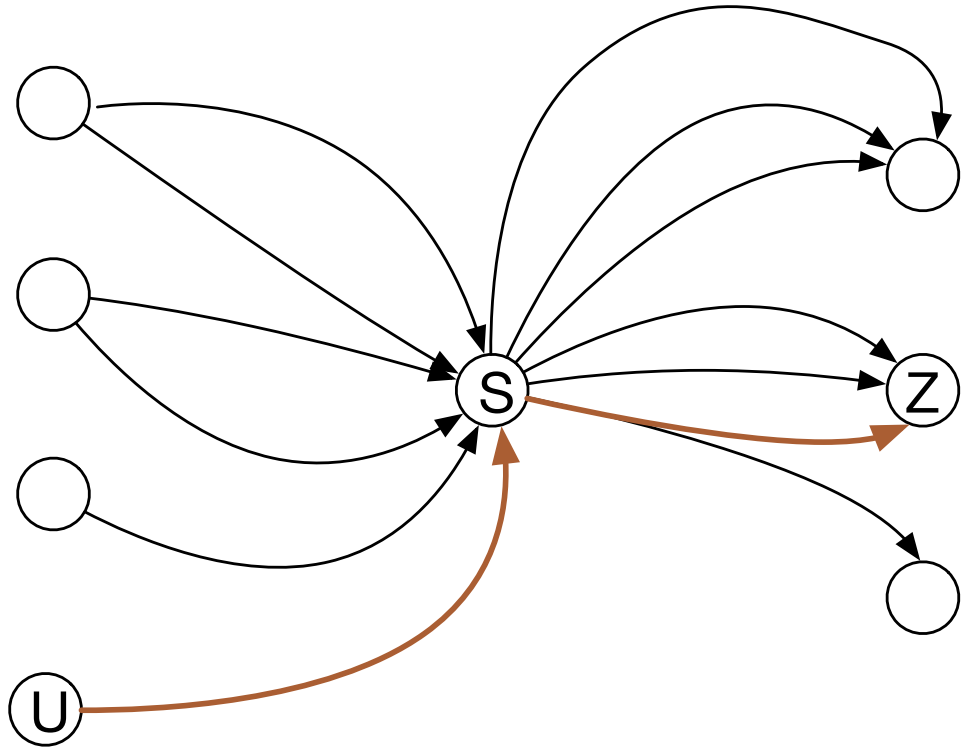
Set $T \gg \frac{1}{1-\gamma}$. For each state s_t visited, and for a learning rate α,

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

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.

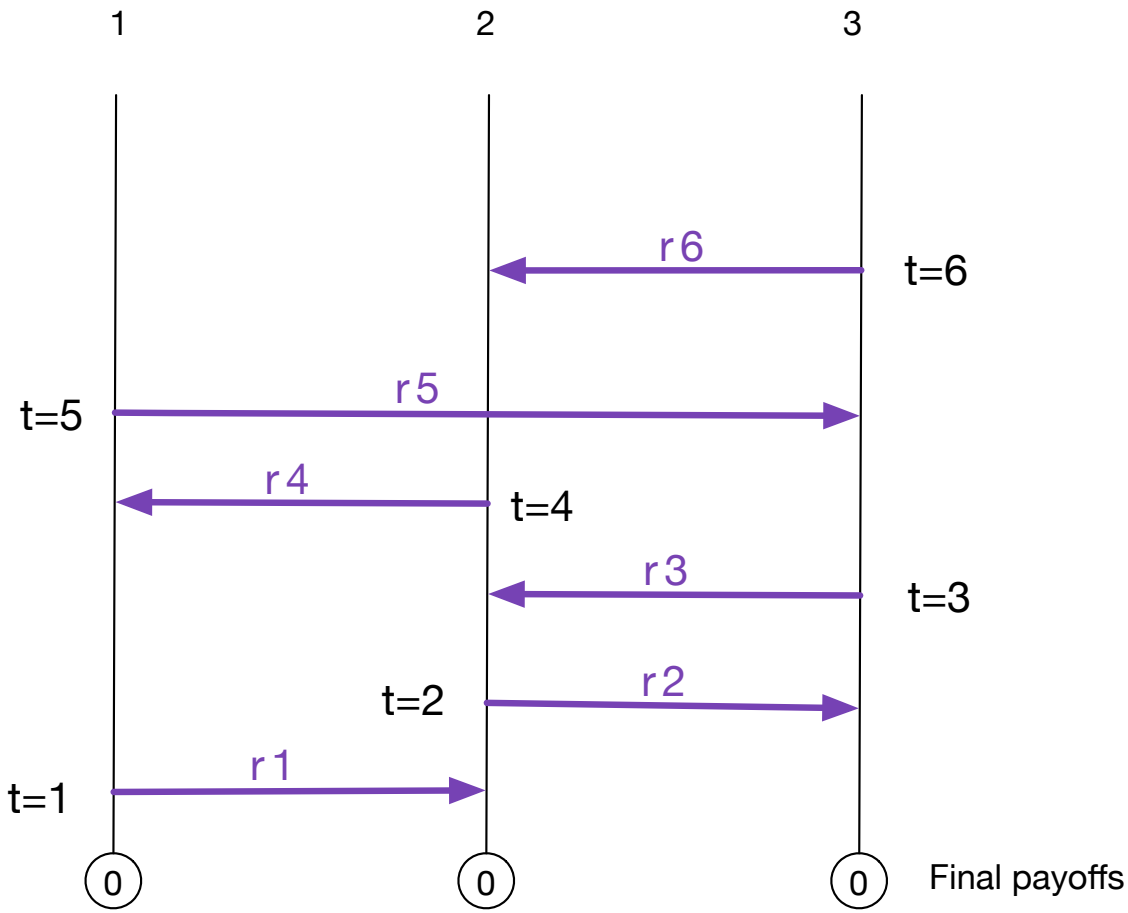
RL part 2: some critical reflection

Chris Watkins

Department of Computer Science
Royal Holloway, University of London

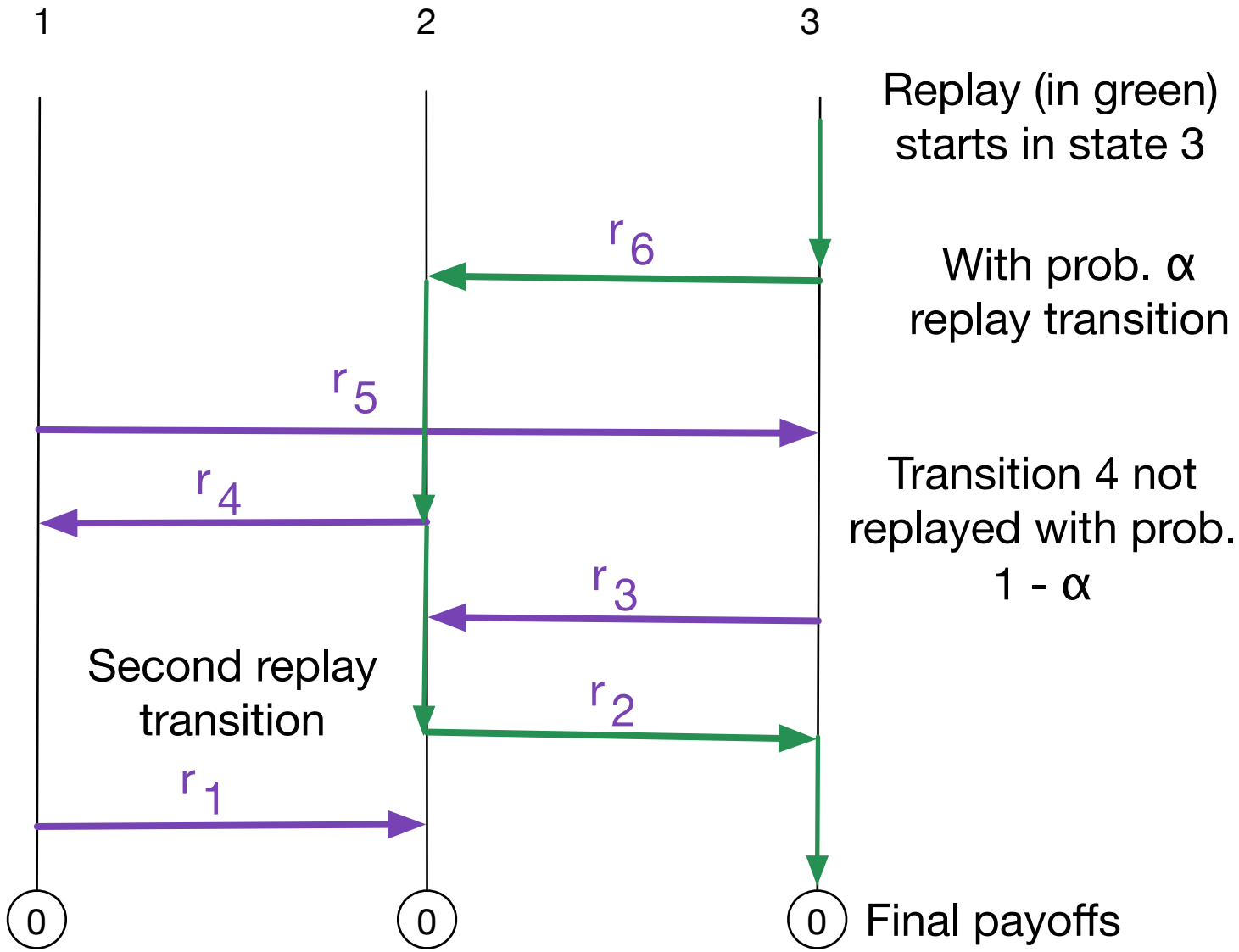
November 25, 2015

Replay process: exact values of replay process are equal to TD estimates of values of actual process



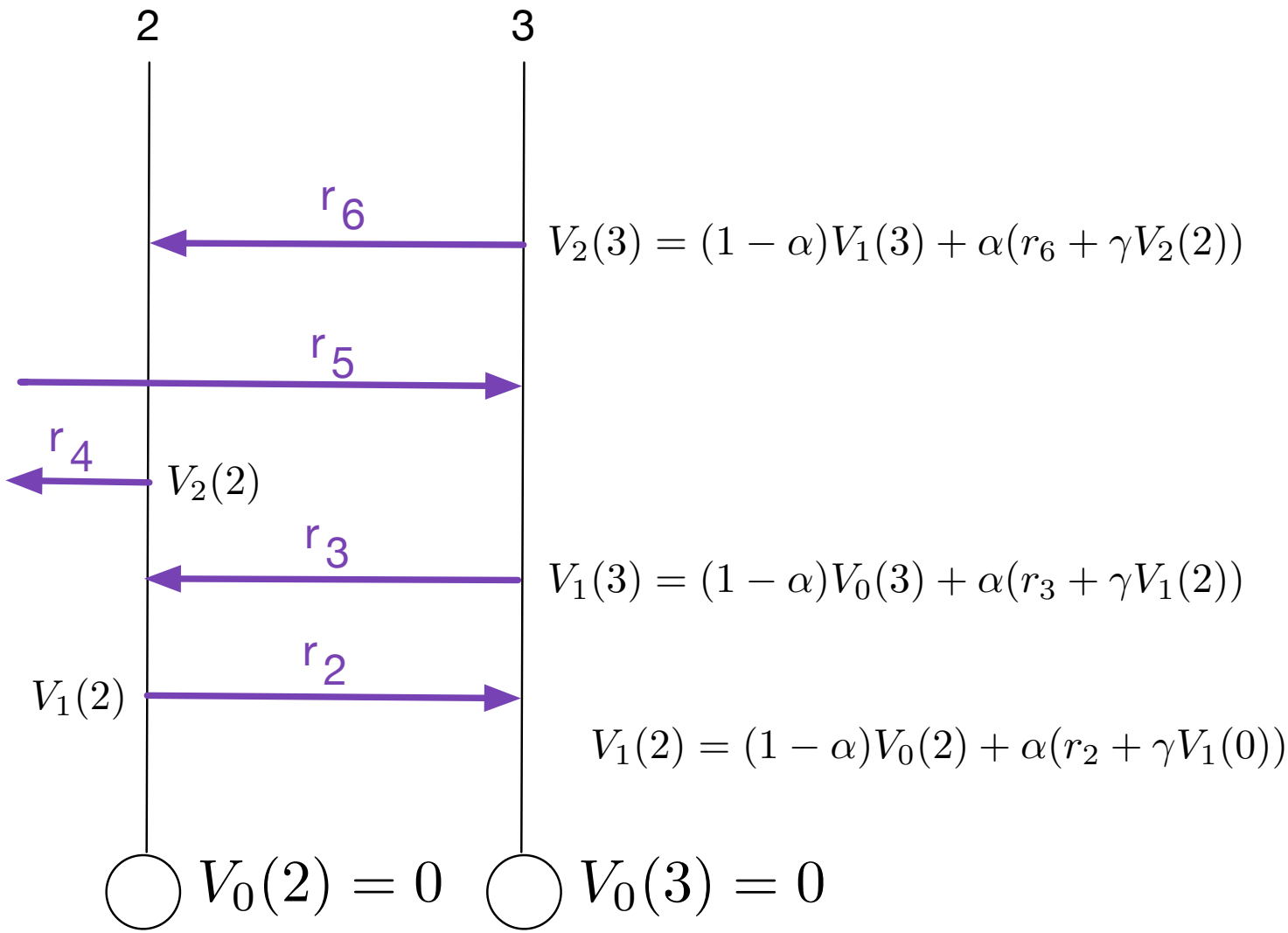
Shows 7 state-transitions and rewards, in a 3 state MDP. Replay process is built from bottom, and replayed from top.

Replay process: example of replay sequence



Return of this replay = $r_6 + \gamma\, r_2$

Values of replay process states



Each stored transition is replayed with probability α

Downward transitions have no discount factor.

Replay process: immediate remarks

- The values of states in the replay process are exactly equal to the TD(0) estimated values of corresponding states in the observed process.
- For small enough α, and with sufficiently many TD(0) updates from each state, the values in the replay process will approach the true values of the observed process.
- Observed transitions can be replayed many times: in the limit of many replays, state values converge to the value function of the maximum likelihood MRP, given the observations.
- Rarely visited states should have higher α, or (better) their transitions replayed more often.
- Stored sequences of actions should be replayed in reverse order.
- Off-policy TD(0) estimation by re-weighting observed transitions

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!

Telescoping of TD errors

$$
\begin{aligned}
TD(1)(s_0) - V(s_0) &= r_0 + \gamma r_1 + \cdots \\
&= -V(s_0) + r_0 + \gamma V(s_1) + \\
&\quad \gamma(r_1 + \gamma V(s_2) - V(s_1)) \\
&\quad \gamma^2(r_2 + \gamma V(s_3) - V(s_2)) \\
&\quad \vdots \\
&= \delta_0 + \gamma\delta_1 + \gamma^2\delta_2 + \cdots
\end{aligned}
$$

Hence the TD(1) error arrives incrementally in the δ_t.

TD(λ)

As a compromise between TD(1) (full reward sequence) and TD(0) (one step) updates, there is a convenient recursion called TD(λ), for $0 \leq \lambda \leq 1$.

The 'accumulating traces' update uses an 'eligibility trace' $z_t(i)$, defined for each state i at each time t. $z_0(i)$ is zero for all i:

$$\begin{aligned} \delta_t &= r_t + \gamma V_t(s_{t+1}) - V_t(s_t) \\ z_t(i) &= [s_t = i] + \gamma\lambda z_{t-1}(i) \\ V_{t+1}(i) &= V_t(i) + \alpha\delta_t z_t(i) \end{aligned}$$

Q-learning of control

An agent in a MDP maintains a table of Q values, which need not (at first) be consistent with any policy.

When agent performs a in state s, and receives r and transitions to s', it is tempting to update $Q(s, a)$ by:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_b Q(s', b))$$

This is a stochastic, partial value-iteration update.

It is possible to prove convergence by stochastic approximation arguments,
but can we devise a suitable replay process which makes convergence obvious?

Replay process for Q-learning

Suppose that Q-learning updates are carried out for a set of $\langle s, a, s', r \rangle$ experiences.

We construct a *replay MDP* using the $\langle s, a, s', r \rangle$ data.

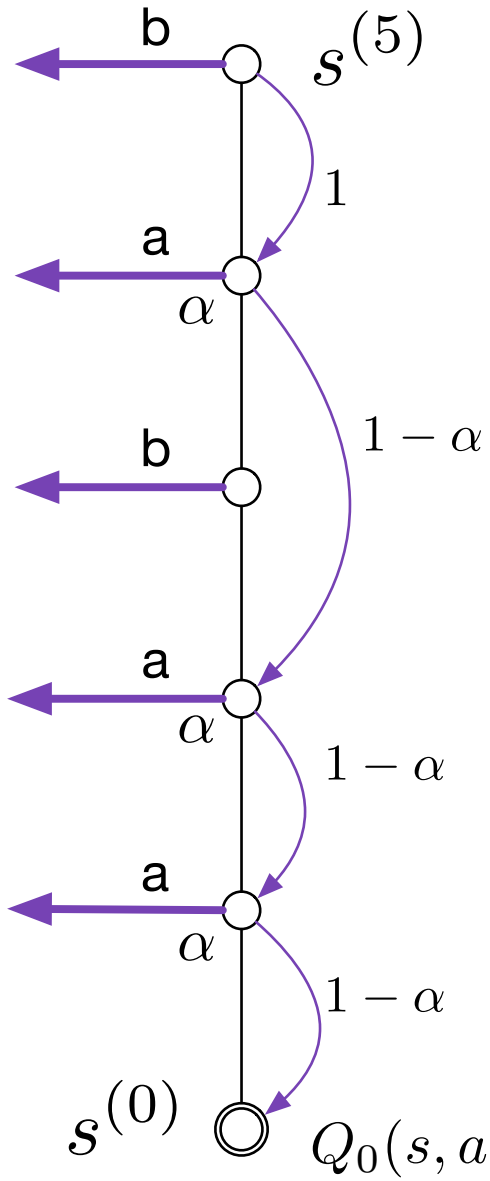
If Q values for s were updated 5 times using the data, the replay MDP contains states $s^{(0)}, s^{(1)}, \ldots, s^{(5)}$.

The *optimal* Q values of $s^{(k)}$ in the replay MDP are **equal** to the *estimated* Q values of the learner after the kth Q learning update in the real MDP.

$$Q_{Real} \;=\; Q^*_{Replay} \;\approx\; Q^*_{Real}$$

$Q^*_{Replay} \approx Q^*_{Real}$ if there are sufficiently many Q updates of all state-action pairs in the MDP, with sufficiently small learning factors α.

Replay process for Q-learning



To perform action a in state s(5):

Transition (with no discount) to most recent performance of a in s;

REPEAT

With probability α replay this performance, else transition with no discount to next most recent performance.

UNTIL a replay is made, or final payoff reached.

Some properties of Q-learning

- Both TD(0) and Q-learning have low computational requirements: are they 'entry-level' associative learning for simple organisms?
- In principle, needs event-memory only for one time-step, but can optimise behaviour for a time-horizon of $\frac{1}{1-\gamma}$
- Constructs no world-model: it samples the world instead.
- Can use replay-memory: a store of past episodes, not ordered in time.
- Off-policy: allows construction of optimal policy while exploring with sub-optimal actions.
- Works better for frequently visited states than for rarely visited states: learning to approach good states may work better than learning to avoid bad states.
- Large-scale implementation possible with a large collection of stored episodes.

What has been achieved?

For *finite* state-spaces and short time horizons, we have:

- solved the problem of preparatory actions
- developed a range of tabular associative learning methods for finding a policy with optimal return
  - Model-based methods based on learning $P(a)$, and several possible modes of calculation.
  - Model-free methods for learning V^*, π^*, and/or Q^* directly from experience.

Computational model of operant reinforcement learning that is more coherent than the previous theory.
General methods of associative learning and control for *small* problems.

The curse of dimensionality

Tabular algorithms feasible only for **very small** problems.

In most practical cases, size of state space is given as number of dimensions, or number of features; the **number of states** is then **exponential** in the number of dimensions/features.

Exact dynamic programming using tables of V or Q values is computationally impractical except for low dimensional problems, or problems with special structure.

A research programme: scaling up

Tables of discrete state values are infeasible for large problems.

Idea: **use supervised learning** to approximate some or all of:

- dynamics (state transitions)
- expected rewards
- policy
- value function
- Q, or the *action advantages* $Q - V$

Use RL, modfiying supervised learning function approximators instead of tables of values.

Some major successes

- Backgammon (TDGammon, by Tesauro, 1995)
- Helicopter manoeuvres (Ng et al, 2006)
- Chess (Knightcap, by Bartlett et al, 2000)
- Motion planning (PILCO, by Deisenroth et al, 2011)
- Multiple arcade games (Mnih et al, 2015)

Practical successes have not been easy.

Challenges in using function approximation

Standard challenges of non-stationary supervised learning, and then in addition:

1. Formulation of reward function
2. Finding an initial policy
3. Exploration
4. Approximating π, Q, and V
5. Max-norm, stability, and extrapolation
6. Local maxima in policy-space
7. Hierarchy

Finding an initial policy

In a vast state-space, this may be hard! Human demonstration only gives paths, not a policy.

1. supervised learning of initial policy from human instructor
2. Inverse RL and apprenticeship learning (Ng and Russell 2000, Abbeel and Ng, 2004)
   Induce or learn reward functions that reinforce a learning agent for performance similar to that of a human expert.
3. 'Shaping' with a potential function (Ng 1999)

Shaping with a potential function

In a given MDP, what transformations of the reward function will leave the optimal policy unchanged? [1]

Consider a finite horizon MDP. Define a *potential function* Φ over states, with all terminal states having same potential. Define an artificial reward

$$\phi(s, s') = \Phi(s') - \Phi(s)$$

Adjust the MDP so that $\langle s, a, s', r \rangle$ becomes $\langle s, a, s', r + \phi(s, s') \rangle$.

Starting from state s, the same total potential difference is added along all possible paths to a terminal state.
The optimal policy is unchanged.

[1] Ng, Harada, Russell, *Policy invariance under reward transformations*, ICML 1999

Exploration

Only a tiny region of state-space is ever visited; an even small fraction of *paths* are taken, or *policies* attempted.

- Inducing exploration with over-optimistic initial value estimates is totally infeasible.
- Naive exploration with ϵ-greedy or softmax action choice may produce poor results.
- Need an exploration plan

Some environments may enforce sufficient exploration: games with a chance (backgammon), and adversarial games (backgammon, chess) may force agent to visit sufficiently diverse parts of the state space.

Approximating π, Q, and V

P may be a 'natural' function, derived from a physical system.
R specified by the modeller; may be simple function of dynamics.

π, Q, and V are **derived** from P and R by an RL operator that involves maximisation and recursion. *Not* 'natural' functions.

Policy is typically both discontinuous and multi-valued.

Value may be discontinuous, and typically has discontinuous gradient.
Either side of a gradient discontinuity, value is achieved by different strategies, so may be heterogeneous.

Q, or 'advantages' $Q - V$, are typically discontinuous and poorly scaled.

Supervised learning of π, V, Q may be challenging.

Max-norm, stability, and extrapolation

Supervised learning algorithms do not usually have max-norm guarantees.

Distribution of states visited depends sensitively on current policy, which depends sensitively on current estimated V or Q.

Many possibilities for instability.

Estimation of V by local averaging is stable (though possibly not accurate). (Gordon 1995)

Local maxima in 'policy-space'

According to the policy improvement lemma, there are no 'local optima' in policy space.
If a policy is sub-optimal, then there is always some state where the policy action can be improved, according to the value function.

Unfortunately, in a large problem, we may never visit those interesting states where the policy could be improved !

'Locally optimal' policies are all too real....

Hierarchy

Three types of hierarchy:

1. Options (macro-operators).
2. Fixed hierarchies (lose optimality)
3. Feudal hierarchies

How state-spaces become large

1. **Complex dynamics**: even a simple robot arm has 7 degrees of freedom. Any complex system has many more, and each degree of freedom adds a dimension to the state-space.
2. A robot arm also has a high-dimension **action-space**. This complicates modelling Q, and finding the action with maximal Q. Finding $\arg\max_a Q(s, a)$ may be a hard optimisation problem even if Q is known.
3. **Zealous modelling**: in practice, it is usually better to work with a highly simplified state-space than to attempt to include all information that could possibly be relevant.

How state-spaces become large (2)

4. **Belief state**: Even if the state-space is small, the agent may not know what the current state is. The agent's actual state is then properly described as a **probability distribution over possible states**. The set of possible **states of belief** can be large.

5. **Goal state**: suppose we wish the system to achieve any of a number of goals: one way to tackle this is to regard the goal as part of the state, so that the new state space is the cartesian product **state-space** $\times$ **goal-space**. Few or rare transitions between different goals: goal is effectively a parameter of the policy.

6. **Reward state**: even in a small system with simple dynamics, the rewards may depend on the history in a complex way. Expansion of reward state happens when an agent is trying to accomplish complex goals, even in a simple system.

Example: Asymmetric Travelling Salesman Problem

Given: distances $d(i, j)$ for K cities; asymmetric so $d(i, j) \neq d(j, i)$.

To find: a permutation σ of $1 : K$ such that
$d(\sigma_1, \sigma_2) + \cdots + d(\sigma_{K-1}, \sigma_K) + d(\sigma_K, \sigma_1)$ is minimal.

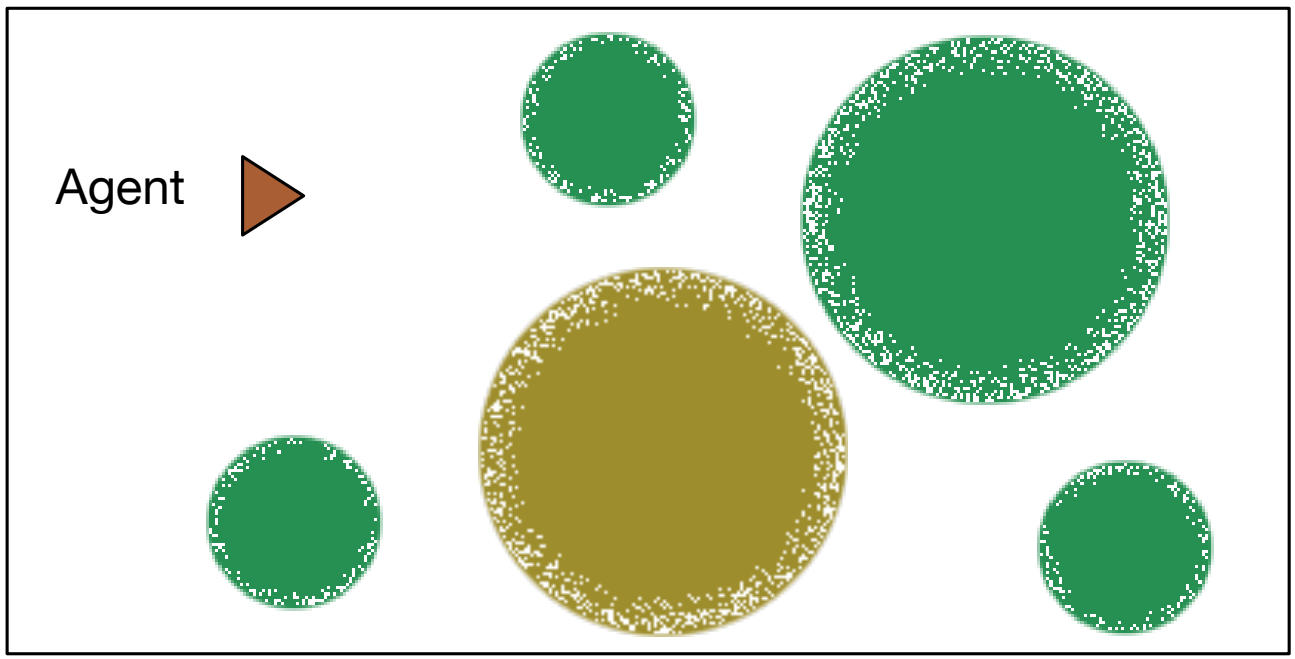
RL formulation as a finite horizon problem:

- w.l.o.g. select city 1 as start state.
- **state** is $\langle$current city, set of cities already visited$\rangle$. Number of states is:
$$N = 1 + (K - 1)2^{K-2}$$
- **actions**: In state $\langle i, S \rangle$, agent can move from i to any state not yet visited.
- **rewards**: In moving from i to j, agent receives $d(i, j)$.
In the $K - 1$ states where all cities have been visited, and agent is at $j \neq 1$, final payoff is $d(j, 1)$.

Although TSP can be formulated as RL, no gain in doing so!

Example: Searching an Area

An agent searches a field for mushrooms: it finds a mushroom only if close to it. What is the state-space?



State includes:

- area already searched: can be a complex shape.
- estimates of mushroom abundance in green and brown areas
- time remaining; level of hunger; distance from home...

Optimisation of Subjective Return?

In RL, the theory we have is for how to optimise expected return from a sequence of immediate rewards.
In some control applications, this is the true aim of the system design: the control costs and payoffs can be adequately expressed as immediate rewards. The RL formalisation then really does describe the problem as it really is.

From point of view of psychology, continual optimisation of a stream of subjective immediate rewards is a strong and implausible theory.
No evidence for this at all !!

A bigger question: where do subjective rewards come from?

Where next?

1. New models: policy optimisation as probabilistic inference, including path integral methods (Kappen, Todorov)
2. ?? New compositional models needed for accumulating knowledge through exploration.
3. Simpler approaches: parametric policy optimisation, cross-entropy method
4. Different models of learning and evolution.

Part 3: Evolution

Chris Watkins

Department of Computer Science
Royal Holloway, University of London

November 25, 2015

Cross-entropy method for TSP

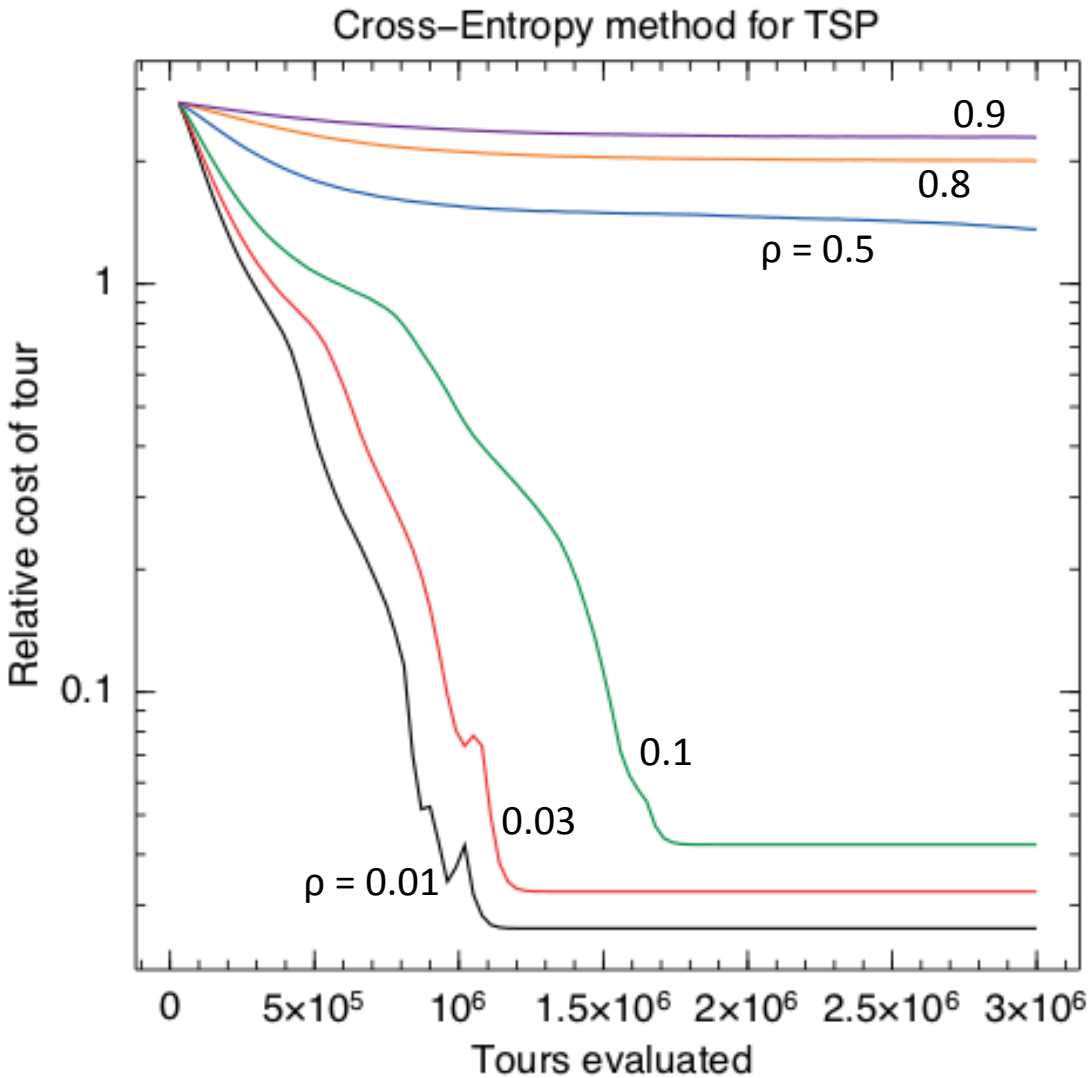
Simple 'genetic style' methods can be effective, such as the following algorithm for 'learning' a near-optimal tour in a TSP.

Initialise: Given: asymmetric distances between C cities
Initialise P, a matrix of transition probabilities between cities (to uniform distributions)

Repeat until convergence:

1. Generate a large number ($\sim 10C^2$) of random tours according to P, starting from city 1, under the constraint that no city may be visited twice.
2. Select the shortest 1% of these tours, and tabulate the transition counts for each city.
3. Adjust P towards the transition counts

53 city asymmetric problem from TSPLib:



Weak selection

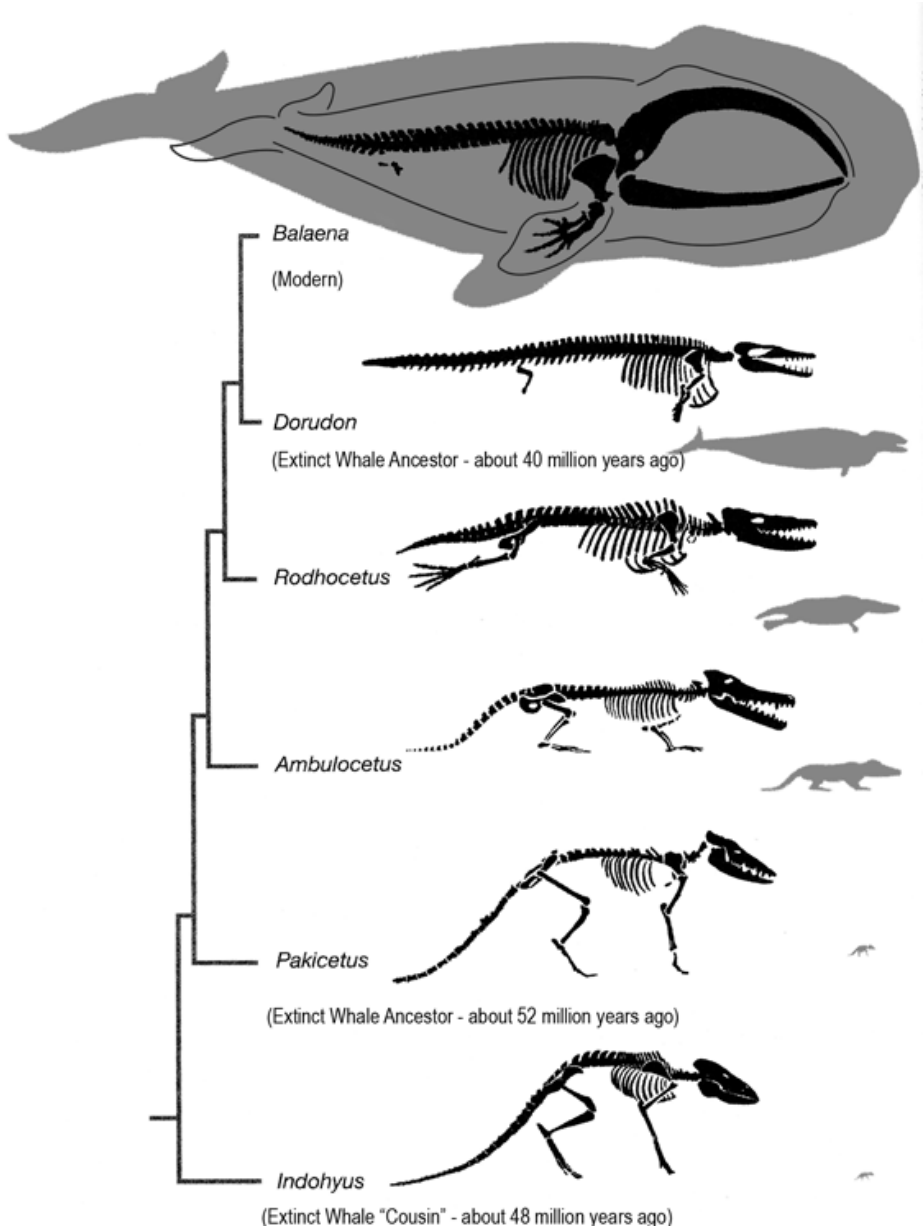
Fraction selected in each generation is ρ

Strong selection

Bowhead whales live up to 200 years



Evolution of whales



Evolution of whales

50 million years ago, ancestors of whales were *Pakicetus*. We only know: [1]

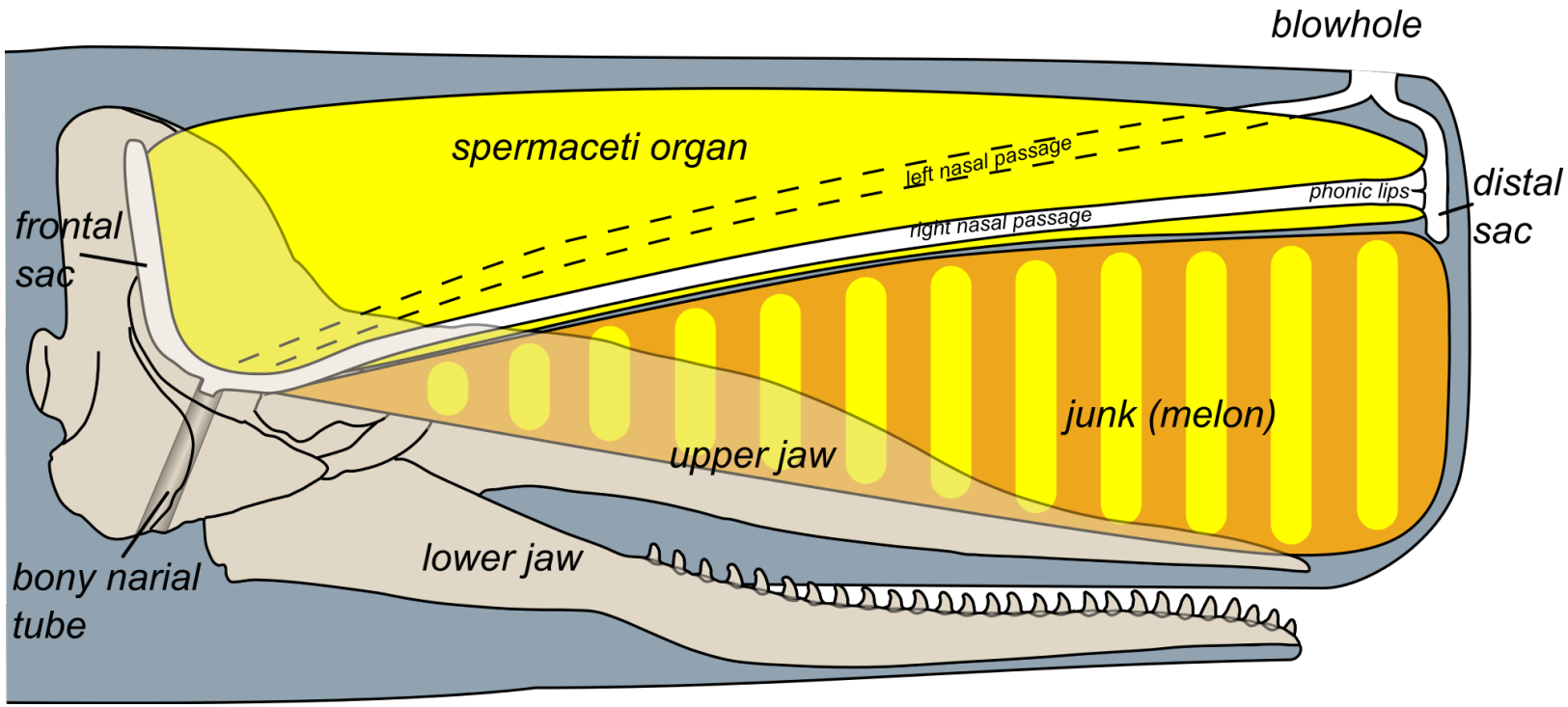
> *Pakicetids are found in northern Pakistan and northwestern India... ...an arid environment with ephemeral streams... the fossils are always recovered from ... river channel deposits...*
>
> *Pakicetids have never been found ... in marine deposits. They were clearly terrestrial or freshwater animals. Their... long thin legs and relatively short hands and feet suggests that they were poor swimmers, and ... it was a very shallow river, probably too shallow for a pakicetid to swim.*
>
> *...it is clear that pakicetids were not fast on land. It appears most likely that pakicetids lived in and near freshwater bodies and that their diet included land animals approaching water for drinking or some freshwater aquatic organisms.*

[1]

Evolution of whales

Modern sperm whales (Physeter) are very different. They dive more than 1000 metres under the sea and hunt, kill, and eat giant squid.

They have a specialised organ (most of their head), which enables them to generate the loudest sound of any animal (230dB). There is another organ which is a sound mirror, that protects their brain.



Evolution of whales: some numbers

Population of sperm whales (*Physeter*) before modern hunting: $\sim 10^6$

Lifespan 60 years; breeds after 9 years; assume 5 years per generation over evolutionary time, so 10^7 generations.

Total number of ancestors from *Pakicetus* to *Physeter* : $\sim 10^{13}$

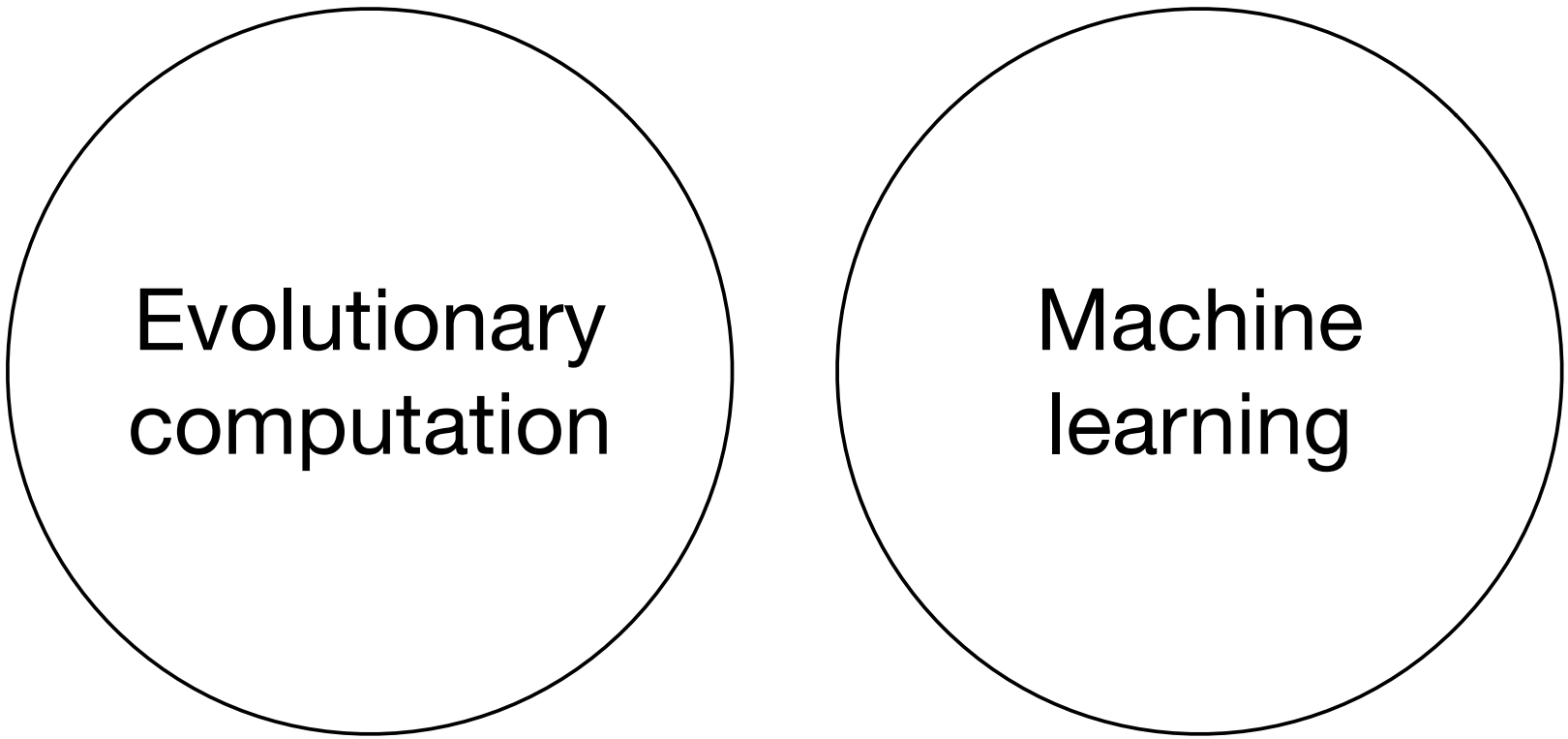
10^{13} individuals is a shockingly small number ! ...and possibly an over-estimate since effective population size may be orders of magnitude smaller.

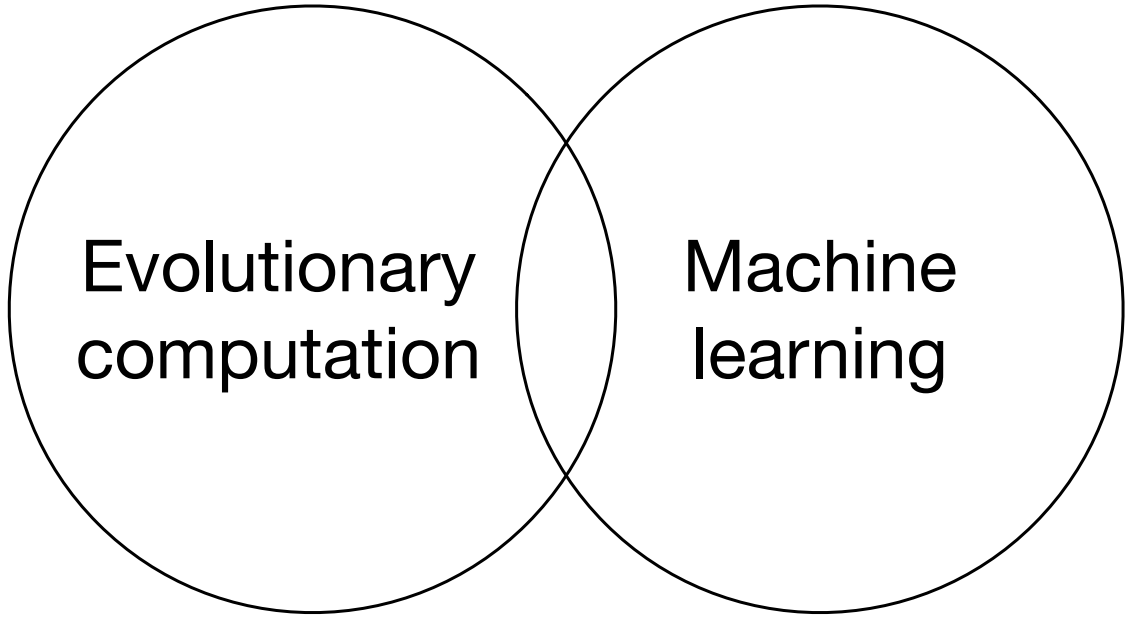
The differences between *Pakicetus* and *Physeter* are difficult to quantify, but include major changes in anatomy, physiology, biochemistry, sensory abilities, and behaviour, from an accidental process of evolution.

Behaviour is an evolved system, genetically specified by reward functions, instincts, and in unknown ways.


Physeter

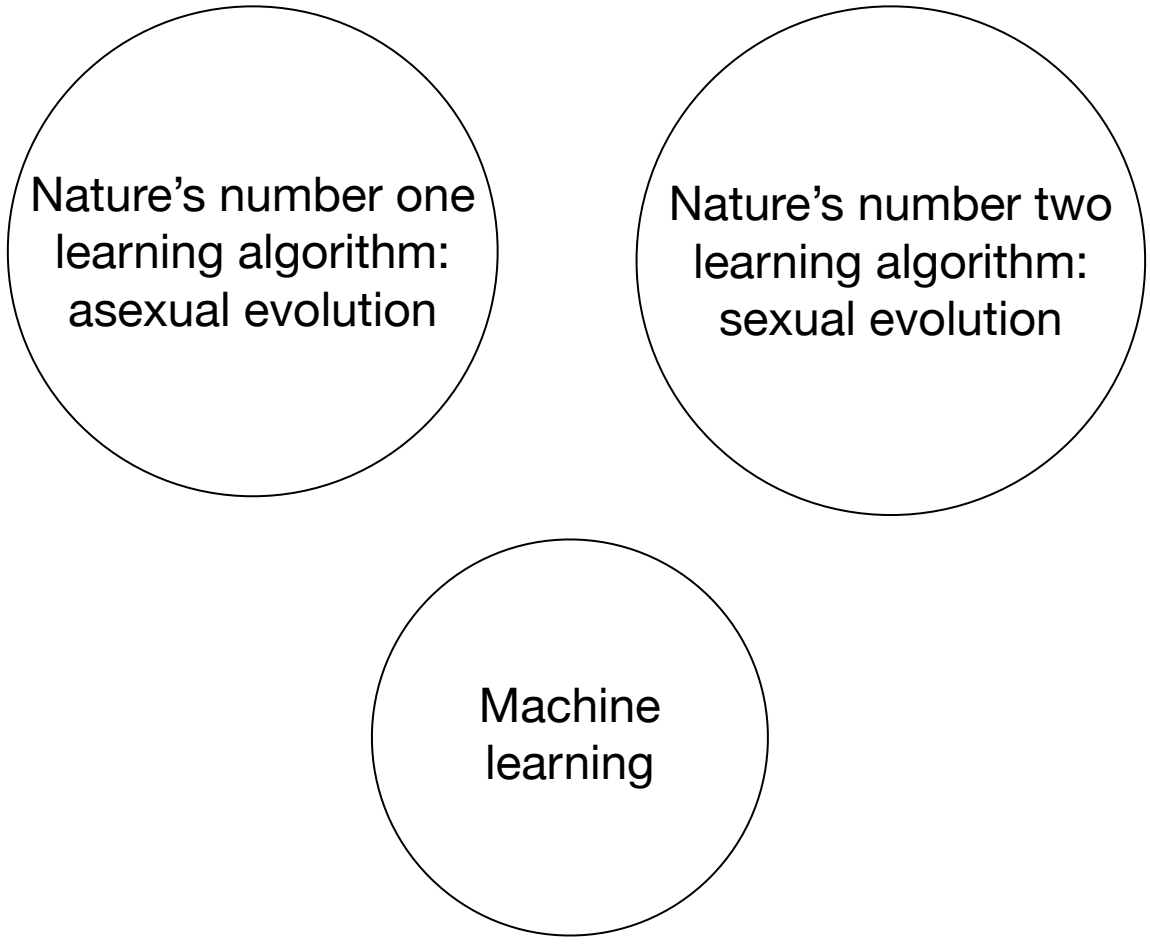
Venn diagram





- Estimation of distribution algorithms (EDA)
  (Pelikan and many others)
- Cross-entropy method (CEM)
  (Rubinstein, Kroese, Mannor 2000 - 2005)
- Evolution of reward functions (Niv, Singh)
- Simple model of evolution and learning
  (Hinton and Nowlan 1987)

Perhaps we should take evolution seriously?



Needed: an 'enabling model' of evolution

The best model of a cat is a cat.

— *Norbert Wiener*

Wiener is wryly pointing out that a model should be as simple as possible, and should contain only what is essential for the problem at hand.

We should discard as many biological details as possible, while keeping the computational essence.

If someone says "But biological detail X is important !", then we can try putting it back in and see if it makes any difference.

Aim:

Can we construct a ‘machine learning’ style model of evolution, which includes *both* genetic evolution *and* individual learning?

Genetic mechanisms of sexual reproduction

At DNA level, the mechanisms of sexual reproduction are well known and tantalisingly simple!

Genetic mechanisms of sexual reproduction were fixed $> 10^9$ years ago, in single-celled protozoa;
since then, spontaneous evolution of advanced representational systems for multicellular anatomy and complex instincts.

Evolution is so robust you can't stop it...

Hypothesis: fair copying is the essence of sexual reproduction

Each child – each new member of the population – is a combination of genetic material copied (with errors) from other members of the population.

The copying is *fair* : all genetic material from all members of the population has an equal chance of being copied into the 'child'

A Markov chain of populations

Suppose we 'breed' individuals under constant conditions. Procedures for breeding, recombination, mutation and for selection are constant.

Then each population depends only on the previous population, so the sequence of populations is a Markov chain.

We will consider a sequence of populations where at each transition just one individual is removed, and just one is added. (In genetic language, this is the Moran model of overlapping populations)

Markov Chains: unique stationary distribution

A Markov chain is specified by its *transition probabilities*

$$T_{ji} = T(i \to j) = P(X_{t+1} = j | X_t = i)$$

There is a *unique*[2] stationary distribution π over the states such that

$$T\pi = \pi$$

In genetic language, the stationary distribution of the Markov chain of populations is the *mutation-selection equilibrium* .

[2] All the Markov chains we consider will be irreducible, aperiodic, and positive recurrent: such chains have a unique *stationary distribution* over states. (Any reasonable model of mutation allows some probability of any genome changing to any other.)

Markov Chains: detailed balance and reversibility

Reversibility

A Markov chain T is *reversible* iff $T\pi = \pi = T^T\pi$

Detailed balance condition

For any two states X, X',

$$\pi(X)T(X \to X') = \pi(X')T(X' \to X)$$

Kolmogorov cycle condition

For all $n > 2$, $X_1, X_2, \ldots, X_n$,

$$T(X_1 \to X_2)T(X_2 \to X_3)\cdots T(X_n \to X_1) = T(X_n \to X_{n-1})T(X_{n-1} \to X_{n-2})\cdots T(X_1 \to X_n)$$

Reversibility $\iff$ Detailed balance $\iff$ Kolmogorov cycle

Markov Chain Monte-Carlo (MCMC) in a nutshell

MCMC is a computational approach for sampling from a known, (typically complicated) probability distribution π.

Given a probability distribution π, *construct* a Markov chain T with stationary distribution π, and run the Markov chain to (eventually) get samples from π.

But how to construct a suitable T ?
Easiest way: define T that satisfies detailed balance for π.

Most common techniques for constructing reversible chains for a specified stationary distribution are:

- Metropolis-Hastings algorithm
- Gibbs sampling

Note that only a 'small' class of Markov chains are reversible: there is no advantage in reversible chains except that it may be easier to characterise the stationary distribution.

Evolution as MCMC

Idea: can we find a computational model of evolution that satisfies detailed balance (and so is reversible) ?

A nice stationary distribution would be:

$$\pi(g_1, \ldots, g_N) = p_B(g_1, \ldots, g_N) f(g_1) \cdots f(g_N)$$

where

- π is an (unnormalised) stationary distribution of a population of N genomes $g_1, \ldots, g_N$
- $p_B(g_1, \ldots, g_N)$ is the 'breeding distribution', the stationary probability of the population with no selection
- $f(g)$ is the 'fitness' of genome g. We require $f(g) > 0$.

Reversibility: a problem

When a reversible chain has reached its stationary distribution, an observer cannot tell if it is running forwards or backwards, because every cycle of state transitions has the same probability in both directions (as in physics).

Any model where a child must have two parents is not reversible

Given parents g_1, g_2, and their child c, the child is more similar to each parent, than the parents are similar to each other. e.g. for genomes of binary values,

$$\text{HammingDistance}(g_1,c) \approx \tfrac{1}{2}\ \text{HammingDistance}(g_1,g_2)$$

Hence in small populations with large genomes, we can identify parents and children, and so identify the direction of time

So natural evolution (and Holland's genetic algorithms) are not reversible Markov chains.

Key idea: Exchangeable Breeding

Generative probability model for a sequence of 'genomes' $g_1, g_2, \ldots$
Breeding only, no selection.

g_1 consists entirely of mutations

g_2 consists of elements copied from g_1, with some mutations

g_3 is elements copied from g_1 or g_2, with fewer mutations

...

Each element of g_N is copied from one of $g_1, \ldots, g_{N-1}$, with few additional mutations...

Generating the sequence $g_1, g_2, \ldots$ does not seem biologically realistic – but *conditionally sampling* g_{N+1} given $g_1, \ldots, g_N$ can be much more plausible.

The sequence $g_1, g_2, \ldots$ needs to be exchangeable

Exchangeable sequences

Definition of exchangeability

A sequence of random variables $g_1, g_2, \ldots$ is *infinitely exchangeable* iff for any N and any permutation σ of $\{1, \ldots, N\}$,

$$p(g_1, \ldots, g_N) = p(g_{\sigma_1}, \ldots, g_{\sigma_N})$$

Which sequences are exchangeable?

de Finetti's theorem

$g_1, g_2, \ldots$ is exchangeable off there is some prior distribution Θ, and a 'hidden parameter' $\theta \sim \Theta$, such that, given knowledge of θ $g_1, g_2, \ldots$ are all i.i.d. samples, distributed according to θ

But de Finetti's theorem does not help us much at this point – we want an example of a plausible generative breeding distribution.

Example: Blackwell-MacQueen urn model

Pick a 'mutation distribution' H. A new genetic element can be sampled from H, independently of other mutations.

Pick a 'concentration parameter' $\alpha > 0$; this will determine the mutation rate

Generate a sequence $\theta_1, \theta_2, \ldots$ by:

- $\theta_1 \sim H$
- With prob. $\frac{1}{1+\alpha}$, $\theta_2 = \theta_1$, else w.p. $\frac{\alpha}{1+\alpha}$, $\theta_2 \sim H$
- $\ldots$
- With prob. $\frac{n-1}{n-1+\alpha}$, θ_n is randomly chosen (copied) from a uniform random choice of $\theta_1, \ldots, \theta_{n-1}$, else with prob. $\frac{\alpha}{n-1+\alpha}$, there is a new mutation $\theta_n \sim H$

This exchangeable sequence is well known in machine learning: it samples from the predictive distribution of a Dirichlet process $DP(H, \alpha)$

Example: Cartesian product of DPs

Each genome $g_i = (\theta_{i1}, \ldots, \theta_{iL})$

Generated with L independent Blackwell-MacQueen urn processes:

For each j, $1 \leq j \leq L$,

the sequence $\theta_{1j}, \theta_{2j}, \ldots, \theta_{Nj}$ is a sample from the jth urn process.

Sequence of genomes $g_1, g_2, \ldots, g_N$ is exchangeable, and is a sample from a Cartesian product of L Dirichlet processes.

This is the *simplest* plausible exchangeable breeding model for sexual reproduction.
Many elaborations possible...

Examples of possible breeding distributions

There is a rich and growing family of highly expressive non-parametric distributions, which achieve exchangeablility through random copying. Dirichlet process is the simplest.

Can construct elegant exchangeable distributions of networks, and exchangeable sequences of 'machines' with shared components...

Exchangeable breeding with tournaments (EBT)

Fitness function f, s.t. for all genomes g, $f(g) > 0$.
Current population of N genomes $g_1, \ldots, g_N$, with fitnesses $f_1, \ldots, f_N$.

Repeat forever:

1. Breed g_{N+1} by sampling from p_B conditional on the current population.

$$g_{N+1} \sim p_B(\cdot \mid g_1, \ldots, g_N)$$

2. $f_{N+1} \leftarrow f(g_{N+1})$. Add g_{N+1} into the population.
3. Select one genome i to remove from the population

$$Pr(\text{remove } g_i) = \frac{\frac{1}{f_i}}{\frac{1}{f_1} + \cdots + \frac{1}{f_{N+1}}}$$

The genomes $\{g_1, \ldots, g_{N+1}\} \setminus \{g_i\}$ become the next population of N genomes.

EBT satisfies detailed balance

Let $G = \{g_1, g_2, \ldots, g_{N+1}\}$

To prove: $\pi(G_{\setminus N+1})\,T(G_{\setminus N+1} \to G_{\setminus i}) = \pi(G_{\setminus i})\,T(G_{\setminus i} \to G_{\setminus N+1})$

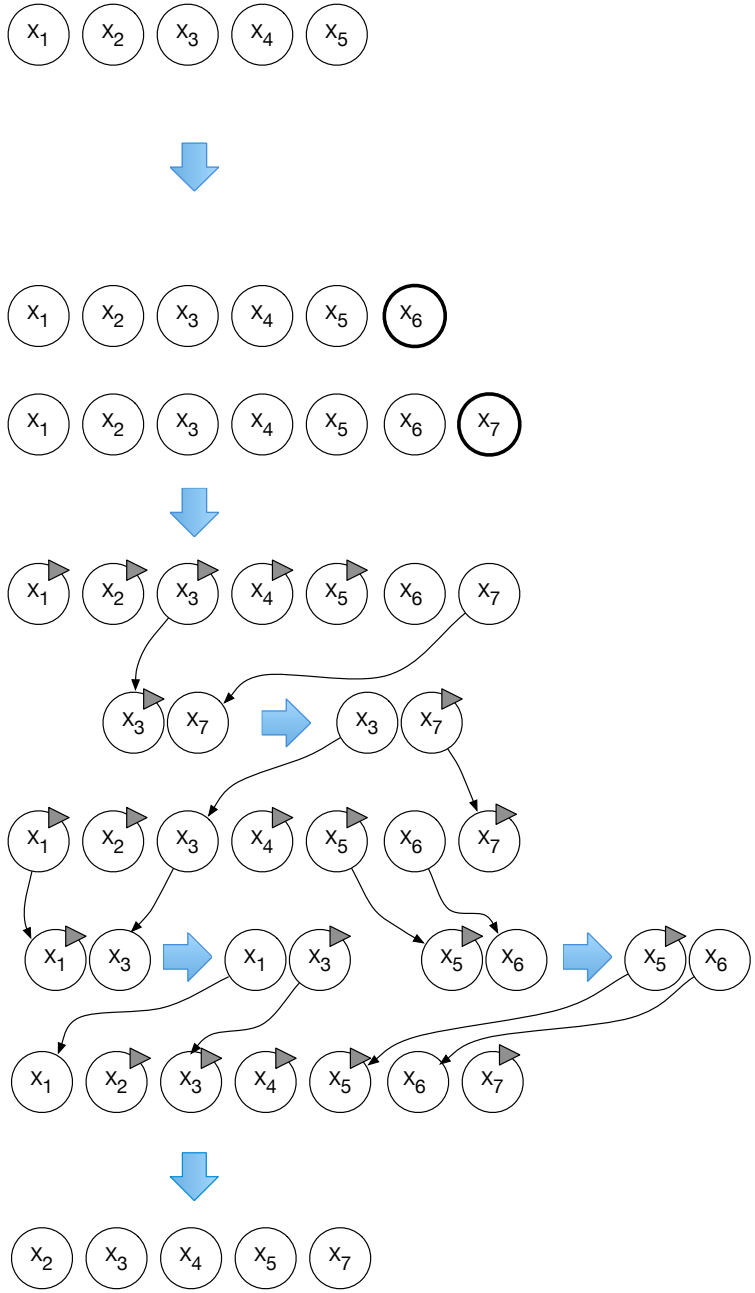
Proof:

$$\pi(G_{\setminus N+1})\,T(G_{\setminus N+1} \to G_{\setminus i}) =$$

$$p_B(G_{\setminus N+1})\ f_1 \cdots f_N\ \ p_B(g_{N+1} \mid G_{\setminus N+1})\ \frac{\frac{1}{f_i}}{\frac{1}{f_1} + \cdots + \frac{1}{f_{N+1}}}$$

$$= p_B(G)\ \ \frac{f_1 \cdots f_{N+1}}{f_i f_{N+1}}\ \ \frac{1}{\frac{1}{f_1} + \cdots + \frac{1}{f_{N+1}}}$$

which is symmetric between g_i and g_{N+1}. *QED*.



Current generation

Breeding phase:

Any number of 'children' exchangeably sampled in sequence.

$$X_6 \sim p_B(\cdot \mid X_1, \ldots, X_5)$$

$$X_7 \sim p_B(\cdot \mid X_1, \ldots, X_6)$$

Selection phase:

Current generation given survival 'tickets' ▶

Any number of 'tournaments' between randomly chosen elements with and without tickets

$$Pr(X_i \text{ wins the ticket against } X_j) = \frac{f(X_i)}{f(X_i) + f(X_j)}$$

Tournaments are shown between X_3 and X_7, then between X_1, X_3, and between X_5, X_6

Next generation are the ticket owners at the end of the selection phase

Role of the population

EBT reduces to Metropolis-Hastings with a population size of 1. Why have a population larger than 1?

Two reasons:

1. Larger population concentrates the conditional breeding distribution $p_B(\cdot \mid g_1, \ldots, g_N)$.
2. Can scale log fitness as $1/N$, thus improving acceptance rate for newly generated individuals.

Stochastically generated environments

If environments are generated from an exchangeable distribution that is independent of the genome, then the environment plays a formally similar role to the genome, in that the stationary distribution for EBT with environments $v_1, \ldots, v_N$ exchangeably sampled from a distribution $P_V(\cdot)$ is:

$$\pi(g_1, \ldots, g_N, v_1, \ldots, v_N) \propto p_B(g_1, \ldots, g_N) p_V(v_1, \ldots, v_N) f(g_1, v_1) \cdots f(g_N, v_N) \quad (1)$$

An intuitive interpretation of this is that each genome is 'born' into its own randomly generated circumstances, or 'environment'; the current population will consist of individuals that are particularly suited to the environments into which they happen to have been born. Indeed, each 'gene' is part of the environment of other genes.

Individual learning

An abstract model of individual learning in different environments is that a genome g in an environment v

1. performs an experiment, observing a result x. The experiment that is performed, and the results obtained, depend on both g and v: let us write $x = \mathit{experiment}(g, v)$.
2. given g and x, the organism develops a post-learning phenotype $\mathit{learn}(g, x)$.
3. the fitness of g in v is then evaluated as

$$f(g, v) = f(\mathit{learn}(g, \mathit{experiment}(g, v)), v)$$

Individual learning

This model captures the notion that an organism:

- obtains experience of its environment;
- the experience depends both on the environment and on its own innate decision making, which depends on the interaction between its genome and the environment;
- the organism then uses this experience to develop its extended phenotype in some way;
- the organism's fitness depends on the interaction of its extended phenotype with its environment.

Minimal assumptions about 'learning'

We do not assume:

- learning is rational in any sense
- organism has subjective utilities or reinforcement

What else?

Exchangeable breeding with tournament selection is closely related to standard methods of Gibbs-sampling for non-parametric Bayesian priors.

Is evolution just Gibbs-sampling ?
What has been left out of the model?

Competition: seems hard to model within this reversible framework. Success for one organism means failure for another. New phenomena?