RL 16: Model-based RL and Multi-Objective Reinforcement Learning

Michael Herrmann

University of Edinburgh, School of Informatics

13/03/2015

Overview

- Complexity of RL
- Model-based
- The Dyna architecture
- MORL

Three sources of error in RL

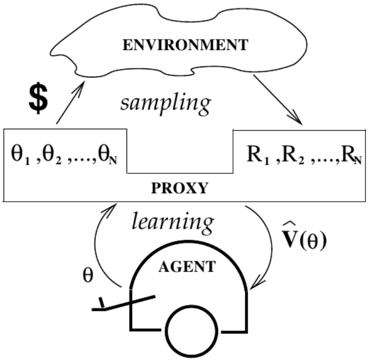
- Misallocation of approximation resources to state space: without knowing the optimal policy one cannot sample from the distribution that it induces on the stochastic system's state space
- Coupling of optimal decisions at each stage: finding the optimal decision rule at a certain stage hinges on knowing the optimal decision rule for future stages
- Inadequate control of generalisation errors: without a model ensemble averages must be approximated from training trajectories

D. Blatt and A. Hero ICAPS Workshop 2006

Types of RL approaches

- Policy search: $\pi : s \to a$
- Value function based: $(s, a) \to V$
  implies policy-based methods by search for the action that maximised value
- Model based $(s, a) \to (s', r)$
  implies value-based methods by solving Bellman equations

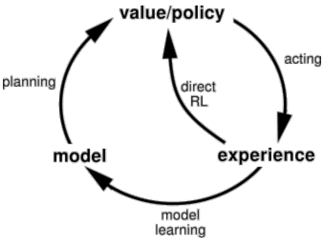
Policy evaluation process: Using a model



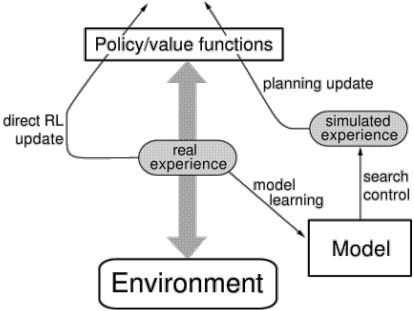
- Sampling process is costly
- Proxy collects the samples from environment and constructs an agent-centric model that predicts the effects of hypothetical agent policies.
- Agent learns by interacting with the proxy.

from Peshkin & Shelton 2001

Model based RL: Dyna



Relationships among learning, planning, and acting.



The general Dyna architecture

Dyna-Q

1. Record state s and select action a
2. Execute action a and record next state s' and reward r,
3. Improve state-action value function using the sample $\langle s, a, r, s' \rangle$
4. Improve world model $M(s, a) \rightarrow (s', r)$
5. Enter planning cycle
   repeat:
   1. Select a random state $\tilde{s}$ and a random action $\tilde{a}$ and
   2. Apply the world model in order to obtain $\tilde{s}'$ and $\tilde{r}$
   3. Improve state-action value function using the sample $\langle \tilde{s}, \tilde{a}, \tilde{r}, \tilde{s}' \rangle$
6. Go to 1

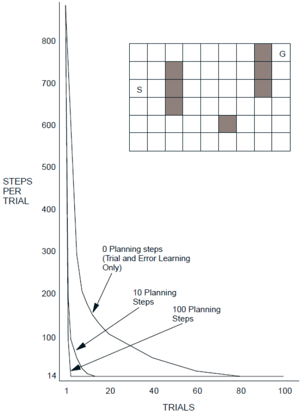
Remarks on Dyna-$\mathcal{Q}$

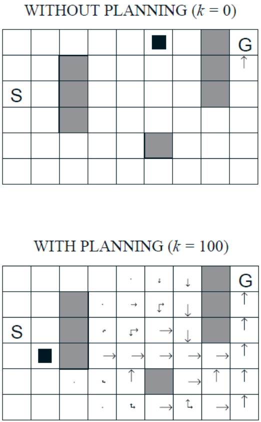
- Dyna-$\mathcal{Q}$ uses $\mathcal{Q}$ learning as a subroutine
- Dyna-$\mathcal{Q}$ uses "dreaming" to obtain a *consistent* value function
- Dyna-$\mathcal{Q}$+ includes an exploration bonus, e.g. $\kappa\sqrt{t(s,a)}$, where $t(s,a)$ is the number of time steps since action a was last executed in state s (in the real world), κ is the exploration strength

$$\mathcal{Q}(s,a) = \mathcal{Q}(s,a) + \eta\left(r + \gamma \max_b \mathcal{Q}(s',b) - \mathcal{Q}(s,a) + \kappa\sqrt{s(s,a)}\right)$$

- Dyna can be used with other algorithms e.g. Dyna-AHC (*adaptive heuristic critic* including a prediction of return, i.e. long-term cumulative reward)

Dyna-AHC: Experiment



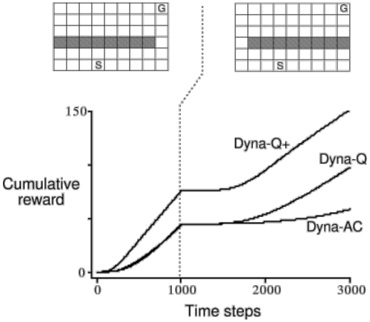


Trial is one trip from start state S to goal state G. Shown are averages over 100 runs.

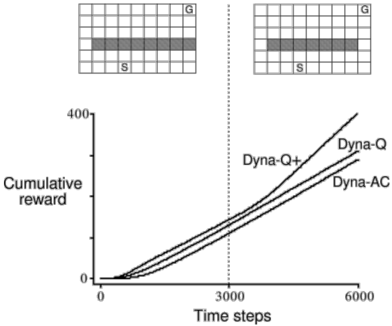
Policies found by the middle of the second trial. Black square is the current location.

R. S. Sutton: Reinforcement Learning Architectures.

When the Model Is Wrong



Blocking task:
Left environment for the first 1000 steps, then right one for the rest.



Shortcut task:
Left environment for the first 3000 steps, then the right one for the rest.

from the Sutton and Barto book

Using a model

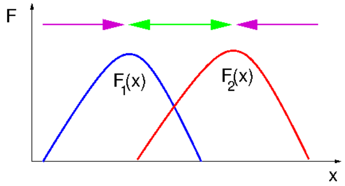
- Large state spaces
  - factorisable transition probabilities
- POMDP with a restricted class of strategies Π
  - chose $\pi \in \Pi$ with maximal return
- what is *sample complexity*? From supervised learning
  - How many samples are needed to learn a function $f \in \mathcal{F}$ of a certain complexity?
  - e.g. neural network realises $h(x)$ with $h \in \mathcal{H}$ in order to approximate $f(x)$. Assume $|\mathcal{H}| = n$ then typically only $O(\log(n))$ samples are needed to find a good $h(n)$.
  - Since we are choosing from $\mathcal{H}$ the complexity of f does not play a role (if $|\mathcal{H}|$ is small and $|\mathcal{F}|$ is large)
- Assume a simulator (a generative model) of the POMDP
- Find bounds on the required amount of simulated experience

Multiobjective Reinforcement Learning

- RL is sequential decision making under uncertainties based on a scalar evaluation signal
- Defining a single reward signal is often the result of a complex design process. Typically several reward signals are available to the agent.
- How can an agent solve several tasks with different rewards simultaneously?
- Does not annihilate information by summing the rewards (which may not be comparable)
- Does the problem become easier or harder for multiple values?
- Robot example: Reach goal(s), avoid wear, keep track of position, avoid getting to close to a human, avoid running out of energy, help other agent that are met on the way ...
- Two main strategies:
  - Scalar combination of the reward signals
  - Pareto optimisation

Multi-objective Optimisation

Example: A machine is characterised by power and torque. A machine is better if – at equal torque – its power is higher.

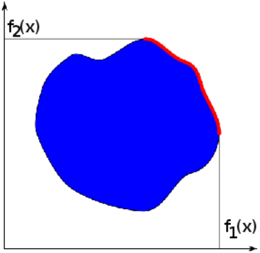


Combination of utility functions, e.g.
$f(x) = |f_1(x)|^\alpha + |f_2(x)|^\alpha$
$f(x) = \alpha f_1(x) + (1-\alpha) f_2(x)$
How to set α?

If α is not implied by the problem, any value in between the two maxima is equally good.

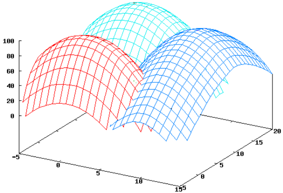
If a comparison between the two quantities is not possible, a set of solutions should be considered as optimal (Pareto-optimal).

How to optimise one criterion without loosing on other criteria?

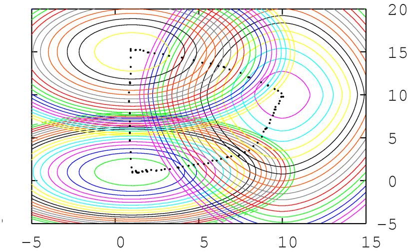


x^* is Pareto optimal for a class of fitness functions $\{f_i\}$ if there exists **no** $x \neq x^*$ with $f_i(x) \geq f_i(x^*)$ for all i

or, equivalently, x^* is not **dominated** by any other x : $\sim\exists x \succ x^*$ (more specifically $\sim\exists x \succ_{\{f_i\}} x^*$)



Example with three fitness functions



Same example: Pareto area spanned by maxima in a shape-dependent way

Multi-objective RL

Two strategies: ... and questions

- Scalarised approach: Find a single policy that optimises a combinations of the rewards
  - Which reward combination is preferable at which state?
  - Although a weighted sum of rewards might be an option, usually a weighted sum of values is considered to more relevant of the actions choice
- Pareto approach
  - Find multiple policies that cover the Pareto front: Sampling in a high-dimensional case
  - In principle, collective search required for sampling the Pareto set
  - What is a good approximation/representation of the Pareto front?

Relation between Scalarisation and Pareto

- A parametrised combination of multiple reward signals is used with different parameters in different runs to address different points along the Pareto front. The set of all solutions obtained in this way contains the Pareto front (e.g. in case of a non-connected Pareto front also non-Pareto optimal solutions may be found)
- The agent may changes the parametrisation according to progress on each of the goals

Approaches to MORL

MORL Approaches		Basic Principle
Single-policy approaches	The weighted sum approach	A linear weighted sum of Q-values is computed as the synthetic objective function.
	The W-learning approach	Each objective has its own recommendation for action selection and the final decision is based on the objective with the largest value.
	The AHP approach	The analytic hierarchy process (AHP) is employed to derive a synthetic objective function.
	The ranking approach	"Partial policies" are used as the synthetic objective function.
	The geometric approach	A target set satisfying certain geometric conditions in multi-dimensional objective space is used as the synthetic objective function.
Multiple-policy approaches	The convex hull approach	Learn optimal value functions or policies for all linear preference settings in the objective space.
	The varying parameter approach	Performing any single-policy algorithm for multiple runs with different parameters, objective threshold values and orderings.

(C. Liu et al., 2013)

$\mathcal{W}$-learning

The Top-$\mathcal{Q}$ algorithm chooses simply

$$a_t = \max_i \mathcal{W}_i = \max_i \max_a \mathcal{Q}_i\left(s_t, a\right)$$

The result depends usually on the scales of the reward signals.

$\mathcal{W}$-learning: Define a principal value function ℓ and choose

$$a_\ell\left(t\right) = \max_a \mathcal{Q}_\ell\left(s\left(t\right), a\right)$$

Calculate $\mathcal{W}$-values by

$$\mathcal{W}_i = \max_a \mathcal{Q}_i\left(s\left(t\right), a\right) - \mathcal{Q}_i\left(s\left(t\right), a_\ell\right)$$

or (to avoid oscillations)

$$\begin{aligned} \mathcal{W}_i\left(s\right) &= \left(1-\alpha\right)\mathcal{W}_i\left(s\right) + \alpha P_i\left(s\right) \\ P_i\left(s\right) &= \max_a \mathcal{Q}_i\left(s, a\right) - \left(r_i + \gamma \max_b \mathcal{Q}_i\left(s', b\right)\right) \end{aligned}$$

set new $\ell = \arg\max_i \mathcal{W}_i$

Approaches to MORL

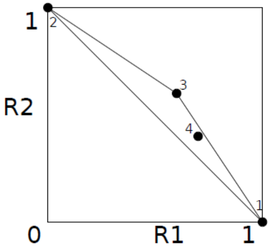
- AHP: Choose action a if is superior for L out of N objectives with a total improvement over the next best action of at least $\Delta\mathcal{Q}$. Combine L and $\Delta\mathcal{Q}$ using a fuzzy system.
- Ranking: Define an ordering of rewards, and check low-priority rewards only if decision is not possible by high-priority rewards.

Advantages of multi-policy approaches

- Agent remains flexible to decide about goals after learning
- Constraints can be expressed by rewards
- "being dominated by" denotes a partial order which is sufficient for many RL approaches
- Non-domination instead of (greedy) maximisation
- Exploration along and across the non-dominated front
- Use several agents (could be represented by the same robot)

Multiple policies

- Convex hull
  Barrett, L., & Narayanan, S. (2008). Learning all optimal policies with multiple criteria. In: 25th ICML, 41-47.
- Varying parameter approach: Finding a Nash-equilibrium of the returns
  C.R. Shelton (2001) Balancing multiple sources of reward in reinforcement learning. NIPS.



convex hull approach

MORL with Policy gradient

- Policy gradient techniques to approximate the Pareto frontier
- How can gradient information be derived from multi-objective sequential decision problems?
- Different MORL approaches based on MO policy gradient
    - radial
    - Pareto following
- see next three slides

Parisi, S., Pirotta, M., Smacchia, N., Bascetta, L., & Restelli, M. (2014) Policy gradient approaches for multi-objective sequential decision making. In: IJCNN, 2323-2330). IEEE.

Slides and source code at: http://home.dei.polimi.it/pirotta

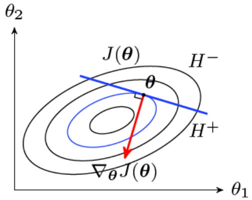
Multi-Objective Policy Gradient (Parisi et al. 2014)

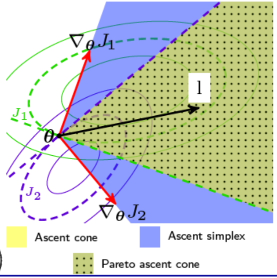
- Half Spaces
- Ascent Cone

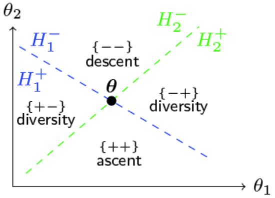
  $C(\boldsymbol{\theta}) = \{\boldsymbol{l} : \boldsymbol{l} \cdot \nabla_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta}) \geq 0\}$
- Ascent Simplex

  $S(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \sum_{i=1}^{q} \lambda_i \nabla_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta})$
- **Pareto–Ascent Cone**

  $S(\boldsymbol{\lambda}, \boldsymbol{\theta}) \cap C(\boldsymbol{\theta})$



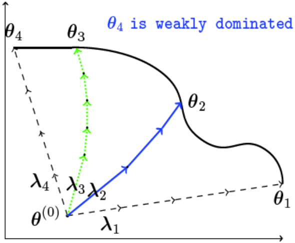


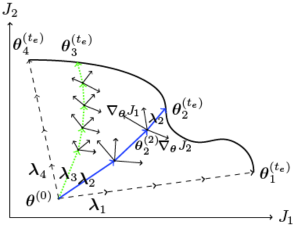


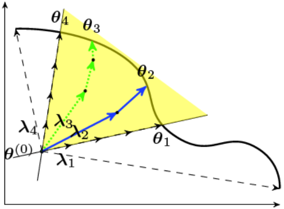
Null Pareto–Ascent Cone $\Rightarrow$ (local) optimal solution

Idea: p gradient ascent searches are performed, each one following a different, *uniformly spaced* direction in the **ascent simplex**

Problem: **weak optimal** solutions





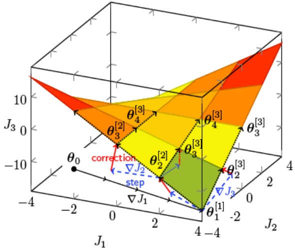


Pareto Following Algorithm (Parisi et al. 2014)

Phase 1: A solution on the Pareto frontier is reached by considering a single objective

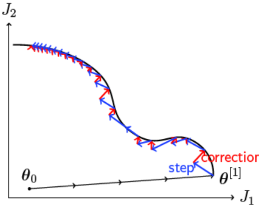
Phase 2: Exploration

- Improvement step: move the solution toward **one** objective at a time
- Correction step: improvement may lead the point outside the frontier. Correction **moves** the point again **on the frontier**



Problems:

- Can reach deterministic policies
- Need to **reintroduce stochasticity** (e.g., based on the entropy)
- Tuning of learning rate



Discussion

- Remain flexible
- Applications, e.g.: Traffic control, Quality of medical service in mobile health care, robot control, network routing, grid computing.
- MARL: In Multi-Agent systems different agents may have different objectives. Different equilibria are possible, differently from the discussed approaches to MORL.

Acknowledgements & References

Liu, C., Xu, X., & Hu, D. (2013). Multiobjective reinforcement learning: A comprehensive overview. *IEEE TA Systems, Man, and Cybern.* **45**:3, 385-398.

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2014). A survey of multi-objective sequential decision-making. *J. Artific. Intellig. Res.* **48**, 67-113.

Parisi, S., Pirotta, M., Smacchia, N., Bascetta, L., & Restelli, M. (2014) Policy gradient approaches for multi-objective sequential decision making. In: IJCNN, 2323-2330). IEEE.

See also:

See also: http://umichrl.pbworks.com/w/page/7597585/Myths of Reinforcement Learning