Reinforcement Learning
1. Introduction

University of Edinburgh, School of Informatics

14/01/2013

Admin

- Lecturer

Michael Herrmann

IPAB, School of Informatics
michael.herrmann@ed (preferred)
Informatics Forum 1.42, 651 7177

- Tutorials
- Mailing list
- Class representative

Admin

- Lectures (<20h): Tuesday and Friday 12:10 - 13:00 (7BSq, LT4)
- Assessment: Homework / Exam 10+10% / 80%
- HW1 (10h): Out 6 Feb, Due 6 March Q-learning: A learning agent in a box-world
- HW2 (10h): Out 6 Mar, Due 27 Mar Continuous-space RL
- Reading/SelfStudy/Solving example problems (32h)
- Tutorials (8h)
- Revision (20h)

Admin

- Tutorials, tentatively:
  - T1 [Bandit problems] – week of 27th Jan
  - T2 [Q-learning] – week of 3rd Feb
  - T3 [MC methods] – week of 10th Feb
  - T4 [TD methods] – week of 17th Feb
  - T5 [POMDP] – week of 24th Feb
  - T6 [continuous RL] – week 3rd Mar
  - T7 [practical aspects] – week 10 of Mar
  - T8 [revision] – week of 17th Mar
- We'll assign questions (combination of pen & paper and computational exercises) – you attempt them before sessions.
- Tutor will discuss and clarify concepts underlying exercises
- Tutorials are not assessed; gain feedback from participation

Admin

Webpage: www.informatics.ed.ac.uk/teaching/courses/rl

- Lecture slides will be uploaded as they become available

Main Readings:

1. R. Sutton and A. Barto, Reinforcement Learning, MIT Press, 1998
2. Csaba Szepesvari: Algorithms for Reinforcement Learning, Morgan & Claypool, 2010 Research papers (later)
3. S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2006 (Chapters 14 – 16)
4. Reinforcement Learning Warehouse http://reinforcementlearning.ai-depot.com/

Background: Mathematics, Probability, Matlab, Machine learning.

What is RL?

- Learning given only percepts (states) and occasional rewards (or punishment)
- Generation and evaluation of a policy i.e. a mapping from states to actions
- A form of active learning
- A microcosm for the entire AI problem
- Neither supervised nor unsupervised

"The use of punishments and rewards can at best be a part of the teaching process" (A. Turing)

Russell and Norvig: AI, Ch.21

Arthur Samuel (1959): Computer Checkers

- **Search tree:** board positions reachable from the current state. Follow paths as indicated by a
- **Scoring function:** based on the position of the board at any given time; tries to measure the chance of winning for each side at the given position.
- Program chooses its move based on a minimax strategy
- **Self-improvement:** Remembering every position it had already seen, along with the terminal value of the reward function. It played thousands of games against itself as another way of learning.
- First to play any board game at a relatively high of level
- The earliest successful machine learning research

wikipedia and Russell and Norvig: AI, Ch.21

More history

- SNARC: Stochastic Neural Analog Reinforcement Calculator (M. Minsky, 1951)
- A. Samuel (1959) Computer Checkers
- Widrow and Hoff (1960) adapted the D. O. Hebb's neural learning rule (1949) for RL: delta rule
- Cart-pole problem (Michie and Chambers, 1968)
- Relation between RL and MDP (P. Werbos, 1977)
- Barto, Sutton, Brouwer (1981) Associative RL
- $\mathcal{Q}$-learning (Watkins, 1989)

Russell and Norvig: AI, Ch.21

Aspects of RL (outlook)

- MAB, MDP, DP, MC, TD(λ), SMDP, POMDP, ...
- Exploration
- Active learning and machine learning
- Structure of state and action spaces
- Continuous domains: Function approximation
- Complexity, optimality, efficiency, numerics
- [Psychology, neuroscience]

Generic Examples

- Motor learning in young children: No teacher. Sensorimotor connection to environment.
- Language acquisition
- Learning to
  - drive a car
  - hold a conversation
  - learning to cook
  - to play games
  - to play a musical instrument
- Problem solving, operations research

Properties of RL learning tasks

- Associativity: Value of an action depends on state
- Active learning: Environment's response affects our subsequent actions
- Delayed reward: We find out the effects of our actions later
- Credit assignment problem: Upon receiving rewards, which actions were responsible for the rewards?

Practical approach to the problem

- Many ways to understand the problem
- Unifying perspective: *Stochastic optimisation over time*
- Given
    - Environment to interact with
    - Goal
- Formulate cost (or *reward*)
- *Objective*: Maximise rewards over time
- The catch: Reward signal not always available, but optimisation is over time (selecting entire paths)
- Let us unpack this through a few application examples . . .

Examples

1. Control
2. Inventory management
3. Chatterbot
4. Playing backgammon, checkers, chess
5. Elevator scheduling
6. Learning to walk in a bipedal robot

(today only 1. - 3.)

Example 1: Control

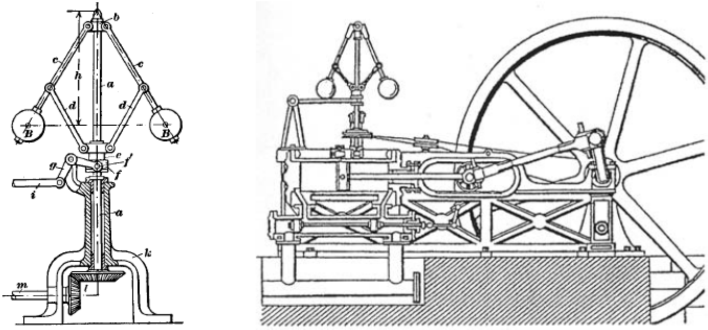


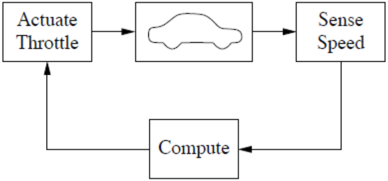
Figure : The centrifugal governor and the steam engine. The centrifugal governor on the left consists of a set of flyballs that spread apart as the speed of the engine increases. The steam engine on the right uses a centrifugal governor (above and to the left of the flywheel) to regulate its speed. (Credit: Machine a Vapeur Horizontale de Philip Taylor [1828].)

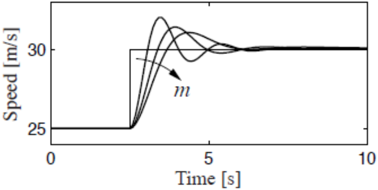
The Notion of *Feedback* Control

Compute corrective actions so as to minimise a measured error

Design involves the following:

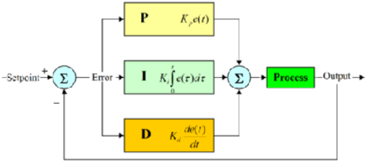
- What is a good policy for determining the corrections?
- What performance specifications are achievable by such systems?





Feedback Control

Proportional-Integral-Derivative Controller Architecture



'Model-free' technique, works reasonably in simple (typically first & second order) systems

- More general: consider feedback architecture $u = -Kx$
- When applied to a linear system, closed-loop dynamics:

$$\dot{x}(t) = (A - BK)\,x(t) = \hat{A}x(t)$$

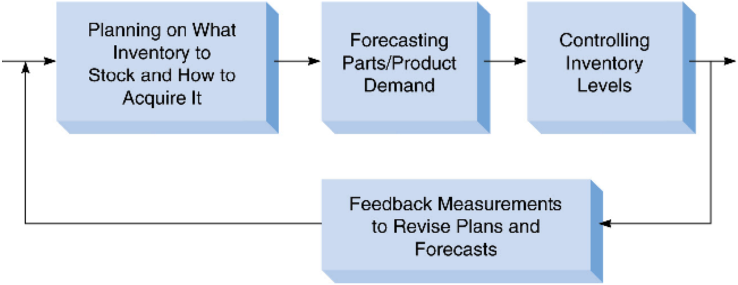
- Using basic linear algebra, you can study dynamic properties
- e.g., choose K to place the eigenvalues and eigenvectors of the closed-loop system

Connection between RL and Control Problems

- RL has close connection to stochastic control
- Main differences seem to arise from what is 'given'
- How to deal with nonlinear systems or system which require adaptation?
- In RL, we emphasise sample-based computation, stochastic approximation

from D. Wolpert

Example 2: Inventory Control


Planning on What Inventory to Stock and How to Acquire It
Forecasting Parts/Product Demand
Controlling Inventory Levels
Feedback Measurements to Revise Plans and Forecasts

- Objective: Minimise total inventory **cost**
- Decisions:
  - How much to order?
  - When to order?

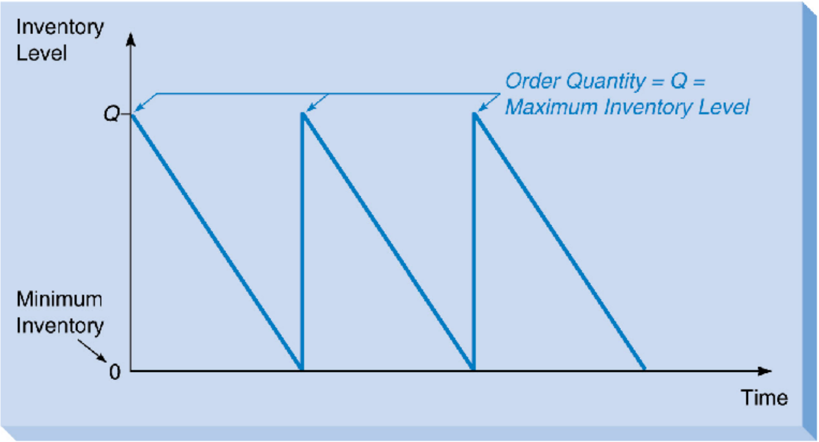
Components of Total Cost

1. Cost of items
2. Cost of ordering
3. Cost of carrying or holding inventory
4. Cost of stockouts
5. Cost of safety stock (extra inventory held to help avoid stockouts)

The Economic Order Quantity Model - How Much to Order?

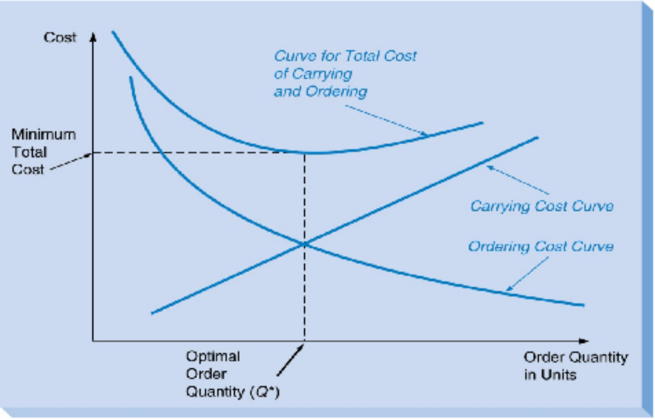
1. Demand is known and constant
2. Lead time is known and constant
3. Receipt of inventory is instantaneous
4. Quantity discounts are not available
5. Variable costs are limited to: ordering cost and carrying (or holding) cost
6. If orders are placed at the right time, stockouts can be avoided

Inventory Level Over Time Based on EOQ Assumptions



Economic order quantity, Ford W. Harris, 1913

EOQ Model Total Cost



At optimal order quantity (Q*): (Carrying cost)' = (Ordering cost)'

$$Q^* = \sqrt{\frac{2DC_o}{C_h}}$$

D: demand, C_o, C_h: costs

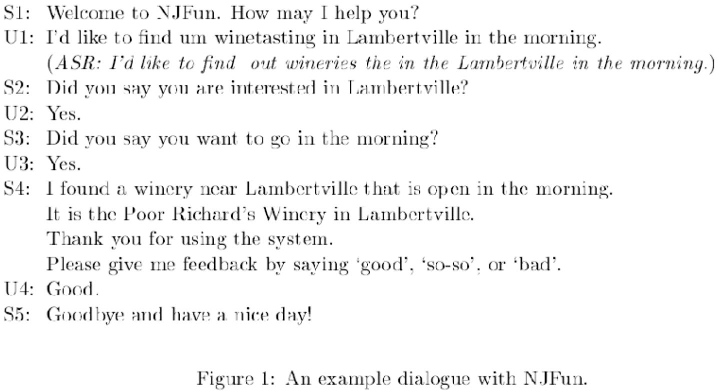
Realistically, how much to order

If these assumptions didn't hold?

- Demand is ~~known~~ and ~~constant~~
- Lead time (latency) is ~~known~~ and ~~constant~~
- Receipt of inventory is ~~instantaneous~~
- Quantity discounts are ~~not~~ available
- ~~Variable costs are limited to~~: ordering cost and carrying (or holding) cost
- If orders are placed at right time, stockouts ~~can be avoided~~

The result may require a more detailed stochastic optimisation.

Example 3: A conversational agent



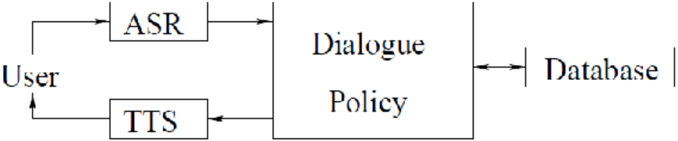
S1: Welcome to NJFun. How may I help you?
U1: I'd like to find um winetasting in Lambertville in the morning.
(ASR: I'd like to find out wineries the in the Lambertville in the morning.)
S2: Did you say you are interested in Lambertville?
U2: Yes.
S3: Did you say you want to go in the morning?
U3: Yes.
S4: I found a winery near Lambertville that is open in the morning.
It is the Poor Richard's Winery in Lambertville.
Thank you for using the system.
Please give me feedback by saying 'good', 'so-so', or 'bad'.
U4: Good.
S5: Goodbye and have a nice day!

Figure 1: An example dialogue with NJFun.

[S. Singh et al., JAIR 2002]

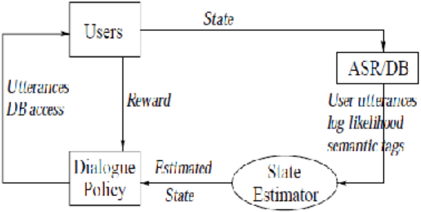
Dialogue management: What is going on?

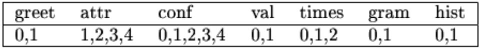
- System is interacting with the user by choosing things to say
- Possible policies for things to say is huge, e.g., 2^{42} in NJFun
- Some questions:
  - What is the model of dynamics?
  - What is being optimised?
  - How much experimentation is possible?



Litman, Diane, et al. "NJFun: a reinforcement learning spoken dialogue system." Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3. Association for Computational Linguistics, 2000.

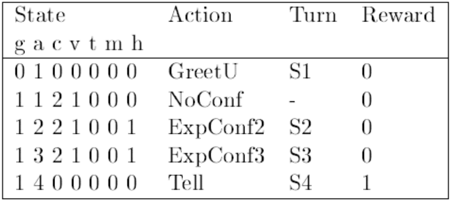
The Dialogue Management Loop





greet	attr	conf	val	times	gram	hist
0,1	1,2,3,4	0,1,2,3,4	0,1	0,1,2	0,1	0,1

State features and values.



State	Action	Turn	Reward
g a c v t m h			
0 1 0 0 0 0 0	GreetU	S1	0
1 1 2 1 0 0 0	NoConf	-	0
1 2 2 1 0 0 1	ExpConf2	S2	0
1 3 2 1 0 0 1	ExpConf3	S3	0
1 4 0 0 0 0 0	Tell	S4	1

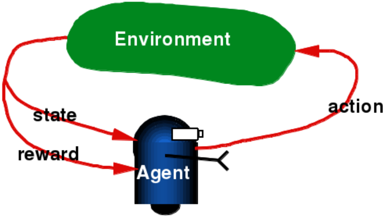
Common Themes in these Examples

- Stochastic Optimisation – make decisions! Over time; may not be immediately obvious how we're doing
- Some notion of cost/reward is implicit in problem – defining this, and constraints to defining this, are key!
- Often, we may need to work with models that can only generate sample traces from experiments

Summary: The Setup for RL

Agent is:

- Temporally situated
- Continual learning and planning
- Objective is to affect the environment – actions and states
- Environment is uncertain, stochastic



Key Features of RL

- Learner is not told which actions to take
- Trial-and-Error search
- Possibility of delayed reward
  - Sacrifice short-term gains for greater long-term gains
- The need to explore and exploit
- Consider the whole problem of a goal-directed agent interacting with an uncertain environment

What is Reinforcement Learning?

- An approach to Artificial Intelligence
- Learning from interaction
- Goal-oriented learning
- Learning about, from, and while interacting with an external environment
- Learning what to do—how to map situations to actions—so as to maximise a numerical reward signal
- Can be thought of as a stochastic optimisation over time

A bit of Maths

Geometric series

$$s_n = \sum_{i=0}^{n} p^n$$

Sum of an infinite geometric series

$$\lim_{n\to\infty} s_n = \frac{1}{1-p}$$

Averages: arithmetic mean, math. expectation, median, mode

Moving average

$$\bar{a}_{i_0} = \frac{1}{N} \sum_{i=0}^{N-1} a_{i_0-i}$$

Weighted (moving) average

$$\bar{a}_{i_0} = \frac{1}{\sum_{i=0}^{N-1} \alpha_i} \sum_{i=0}^{N-1} \alpha_i a_{i_0-i}$$

Exponentially weighted average $\bar{a}_{i_0} = (1-\alpha) \sum_{i=0}^{\infty} \alpha^i a_{i_0-i}$

Acknowledgements

Many slides are adapted from web resources associated with Sutton and Barto's Reinforcement Learning book

... before being used by Dr. Subramanian Ramamoorthy in this course in the last three years.