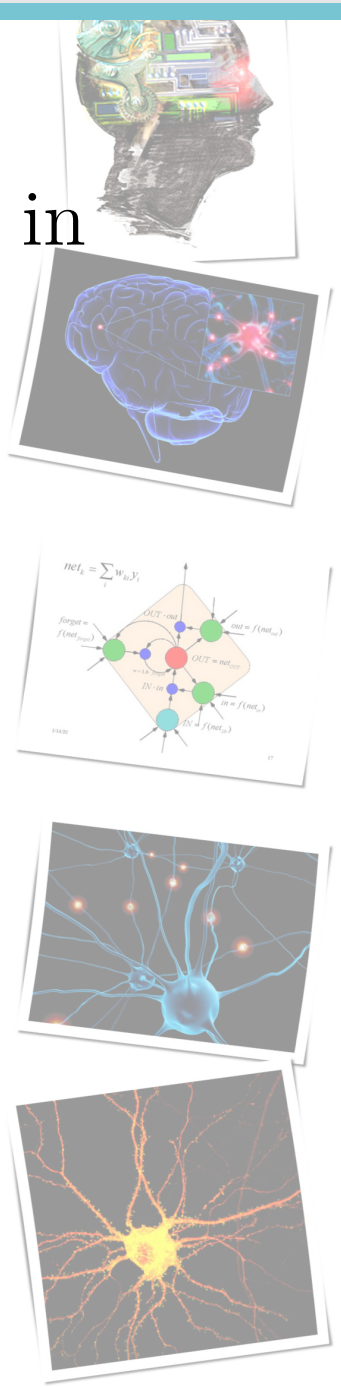
Unifying Learning, Planning and Search in Arbitrary Environments

Spyros Samothrakis
Assistant Director, IADS
University of Essex

#ODSC

September 21, 2018



Introduction

- What follows is a guided tour
  - RL, prescriptive analytics
- Not an introduction
  - I will be introducing some basic concepts, I assume some knowledge on Reinforcement Leaning, tree searches, supervised learning
- An alternative name for the talk could have been why "Deep RL is fundamentally broken and current solutions are unsatisfactory"
  - Supervised learning mostly works out of the box
    - RL doesn't!
- References in most slides

Overview - Analytics

- Descriptive
  - "I'll create high level, compressed views of your data"
  - What is sometimes termed analytics, unsupervised learning
- Predictive
  - "I'll try to predict a possible version of your future".
  - Show me some good customers in your database, I'll try getting you more
  - Supervised learning (mostly)
- Prescriptive
  - "What should I do to achieve certain results?"
  - **Reinforcement Learning, Bandits and Causality**
    - e.g. Send the right sequence of e-mails to engage an audience

---

Chris Wiggins. "Lectures delivered Aug 8-9, 2016 at MLSS.cc (Arequipa, Peru).
https://www.slideshare.net/chrishwiggins/machine-learning-summer-school-2016/75

The MDP

- Defines a universe (?)
  - *States* $s \in \mathcal{S}$, where s is a state in general and $S_t \in \mathcal{S}$ is the (particular) state at time t.
  - *Actions* $a \in \mathcal{A}$, where a is an action in general and $A_t \in \mathcal{A}(S_t)$ is the action at time t, chosen among the available actions in state S_t. If a state has no actions, then it is *terminal* (e.g., endgame positions in games).
  - *Transition probabilities* $p(s'|s,a)$: the probability of moving to state s' when taking action a in state s: $\Pr\{S_{t+1} = s' | S_t = s, A_t = a\}$.
  - *Rewards* $r(s,a,s')$: the expected reward after taking action a in state s and moving to state s', where $R_{t+1} = r(S_t, A_t, S_{t+1})$.
  - The *reward discount rate* $\gamma \in [0,1]$, which decreases the importance of later-received rewards.
- Which is populated by an agent
  - *policy* $\pi(s,a)$
  - *return* $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$.

RL Philosophy

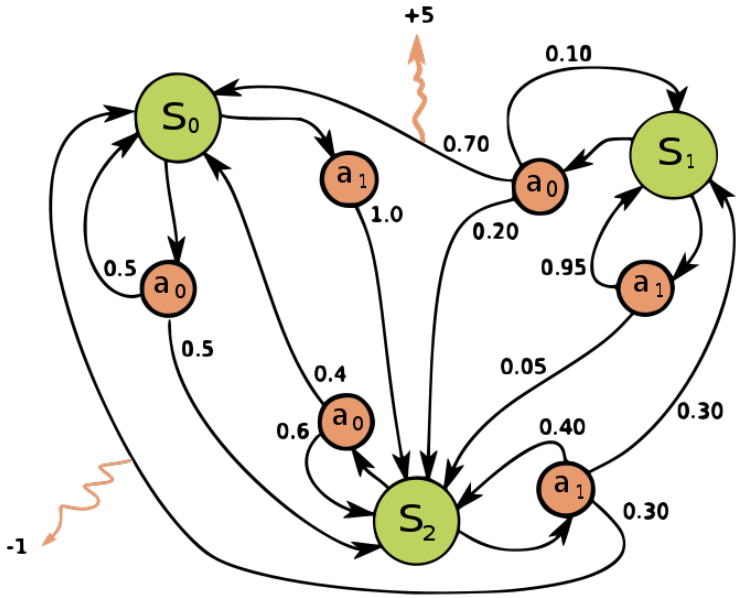
- RL is used quite often as an abstraction that encapsulates the whole of AI
  - There is cruel universe that has left you in a state of no pleasure
  - You need to find a way of escaping that no-pleasure state and go towards happiness
- *Man vs Nature*
  - The **Augustinian** universe
- Assumes a quasi-mad MDP creator
  - Why not just give maximum reward in the initial state the agent is in and just leave it there?
- The model has more to do with Hobbs, liberalism and certain philosophical trends

---

Wiener, Norbert. "The Human Use of Human Beings." (1950).

What is the agent doing?

- Tries to move from low reward states to high-reward states
- *Search* for high-rewarding states
- *Learn* how to get to high rewarding states
- *Plan* how to get to high rewarding states
  - *Learning a model*
  - *Thinking really hard*



Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998

`https://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/Markov_Decision_Process.svg/800px-Markov_Decision_Process.svg.png`

Model-Free RL

- Learn $Q^\pi(s,a)$ and/or $V^\pi(s)$-values and/or $\pi(s,a)$
  - Expectations over returns for some state-action/state, given a certain policy
  - So we are "searching" for good rewards and *learning* V-values and Q-values
  - Q-learning, SARSA, Actor-Critic, Distributed RL
    - Smart use of the sample returns $(s,a), G_t$
- High-reward states are often sparse
  - Hence the need for an active search process (more on this later)

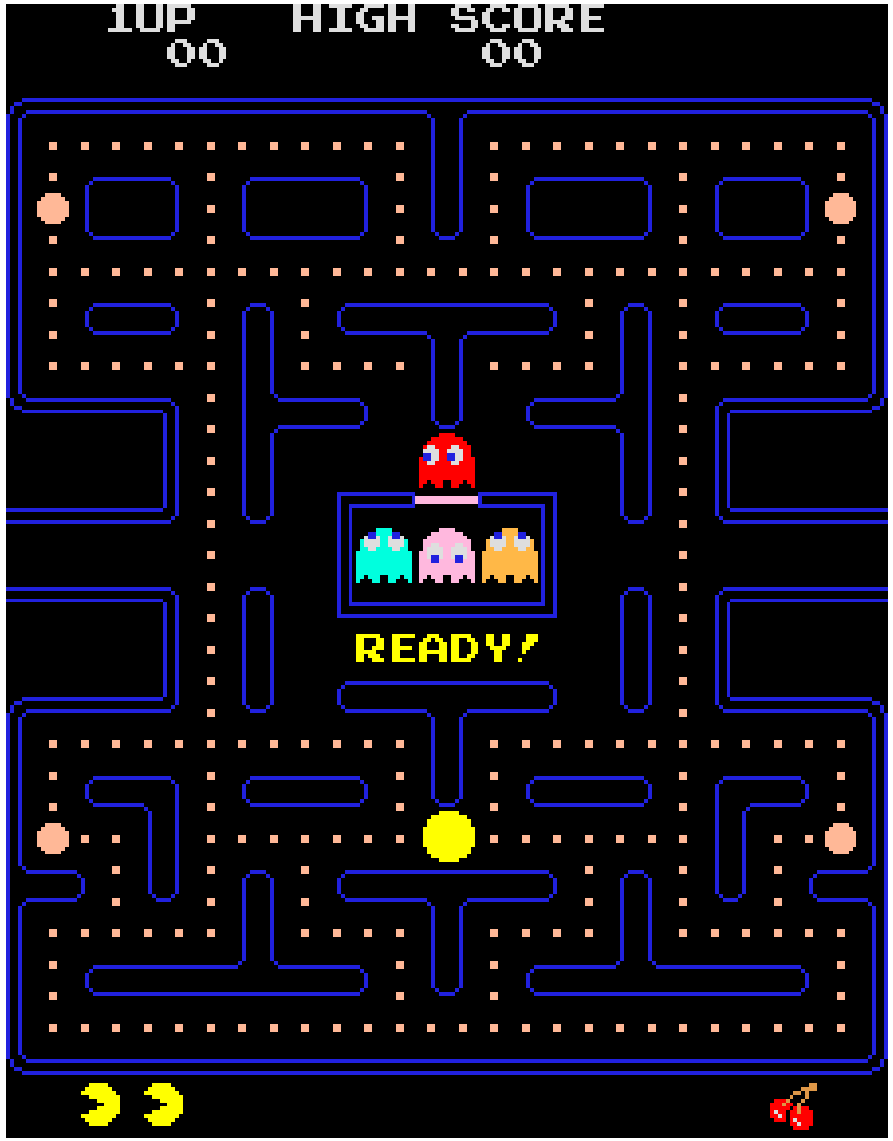
Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

Whiteson, Shimon. "Evolutionary computation for reinforcement learning." Reinforcement learning. Springer, Berlin, Heidelberg, 2012. 325-355.

Real world problems..(?)

Ultimate goal is to drop agent in a simulated universe, agent becomes super-human

- Real world problems almost never have such a clear structure
- States are images, videos, text, sound
- Actions can be words, joystick moves, moving actuators etc
- We need function approximators
  - So all are policies, V/Q values depend on parameters θ of a function approximator



Function Approximators - what are we learning?

- All the above methods work just fine if there is no function approximation
  - Extremely slow, unrealistic process for scaling up RL
  - Only works in toy problems
- Think of it as trying to learn a *regressor* that maps states (and actions) to a Q/V/Policy
- If the function approximator is local, everything should be more or less fine (e.g. a table of values)
- If the function approximator is global, we are in trouble

Local vs global

- Local function approximator examples
  - Tables
  - Radial basis function neural networks
  - N-Tuple networks
  - K-NN
- Global function approximators
  - Neural networks
  - Gradient boosters
  - Random Forests

Most modern ML is about global function approximators

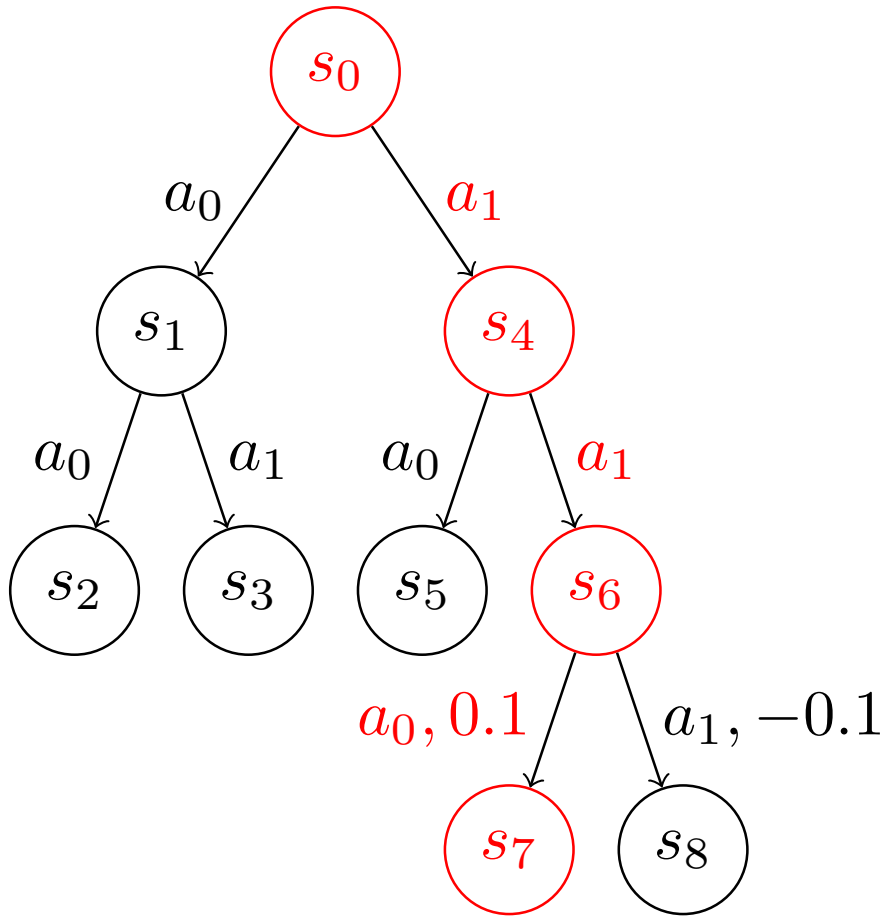
Speculation I - RL problems are really supervised learning problems

- Covariate shift
  - You change the policy, hence your distribution of inputs S, A changes over time!
    - If you throw away past samples, you will forget how to act under bad scenarios
    - You wouldn't even know they actually exist, you forgot about them!
- Concept Shift (or Drift)
  - Your Q-values change as you learn - state Q values are meaningless after a while
- Imbalanced classes (or similar to)
  - Rewards are often sparse - you need to explore

Storkey, Amos. "When training and test sets are different: characterizing learning transfer." Dataset shift in machine learning (2009): 3-28.

Covariate shift (Current state)

- Let's see an example
  - Super-simple MDP
- At some point you will stop seeing suboptimal states
  - Your state/action distribution will change
- So you need to go back and check what those Q-values are
  - Policy cycling
- Might be fixable by tuning



Riedmiller, Martin. "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method." European Conference on Machine Learning. Springer, Berlin, Heidelberg, 2005.

Covariate shift (Research pathway)

- **Detect if you the new stream of (s,a) is different vs your current saved data**
- **Create a classifier that can tell you which stream your data is part of**
- **Many mini-regressors**

Now you have something that looks like a cross between a table and neural-network

Concept Drift/Shift

- Your Q/V values change as your policy changes
  - Feature actions in the chain impact past actions
- You probably need to stick to neural networks, as they are one of the few non-linear regressors you can learn incrementally
- You could use trees if you wanted, but it has to be one of the differentiable versions
- You can't do the easy thing and throw whatever method you have at your data!

Kontschieder, Peter, et al. "Deep neural decision forests." Proceedings of the IEEE international conference on computer vision. 2015.

Good exploration strategies - artificial curiosity

- Core idea: Looking for *Epistemic uncertainty*
  - **If I can't use the past to predict the present, my model is broken**
- Learn forward model - $F(s, a) \rightarrow s'$
  - Can you predict TV noise?
  - *Aleatoric uncertainty*
- Learn inverse model - $I(\phi(s), \phi(s')) \rightarrow a$
  - You can predict with some certainty what action brought you here
- $F(\phi(s), a) \rightarrow \phi(s')$
  - We only keep *epistemic* uncertainty

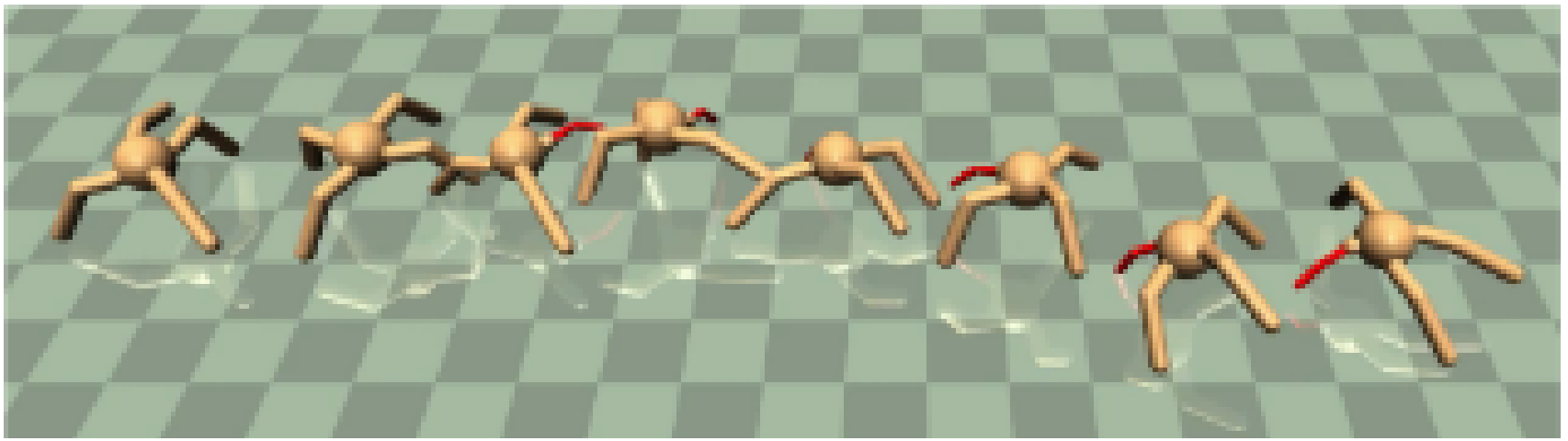
Pathak, Deepak, et al. "Curiosity-driven exploration by self-supervised prediction." International Conference on Machine Learning (ICML). Vol. 2017. 2017.

Model-based and Model-Predictive

- Learn the dynamics/transition function $\Pr\{S_{t+1} = s' | S_t = s, A_t = a\}$
  - No concept drift, covariate shift is still a problem if you are to train online
- You need to learn a model anyway, you might as well use it,
  - Not that easy
  - How many steps forward?
  - Chaotic dynamics if you try recursing
- Monte Carlo Tree Search on the learned model (or if you have a model already)
  - **The original UCT searches identically as an offline on-policy every-visit MC control algorithm that uses UCB1 as the policy**

Vodopivec, Tom, Spyridon Samothrakis, and Branko Ster. "On Monte Carlo tree search and reinforcement learning." Journal of Artificial Intelligence Research 60 (2017): 881-936.

Meta-learning



- Possibly the future (though I am not sold)
- Learning to learn

Clavera, Ignasi, et al. "Learning to Adapt: Meta-Learning for Model-Based Control." arXiv preprint arXiv:1803.11347 (2018).

What is state and where is it located?

- Sufficient statistics that one can act on
  - A collection observations $O_t, O_{t+1}, O_{t+2} + \ldots$.
  - The internal state of a recurrent neural network b_{t+n}
  - Images, Text, human engineered features ϕ_t
- In the archetypal RL literature states and actions are discrete entities
  - Obviously this doesn't scale in the real world
  - Not even when a specific encoding of states exits (e.g. think Chess)
  - Thus the need for function approximators
    - Almost always some kind of neural network

Video Games

- The prime example of RL universes
- States are mostly a bunch of frame stuck together
- Map naturally to RL semantics
- A very rich history of benchmarks and competitions (Pac-man, Mario, Atari games, GVG-AI etc)
  - Minor differences in implementations quite often make comparisons impossible

---

Perez-Liebana, Diego, et al. "The 2014 general video game playing competition." IEEE Transactions on Computational Intelligence and AI in Games 8.3 (2016): 229-243.

Bellemare, Marc G., et al. "The arcade learning environment: An evaluation platform for general agents." Journal of Artificial Intelligence Research 47 (2013): 253-279.

Kempka, Michał, et al. "Vizdoom: A doom-based ai research platform for visual reinforcement learning." Computational Intelligence and Games (CIG), 2016 IEEE Conference on. IEEE, 2016.

Text Based RL

- What if states and actions where text?
  - Most common form of data
- Action space is every possible word
- Actions needs to be discovered!
- Also partial observability - what is your state?

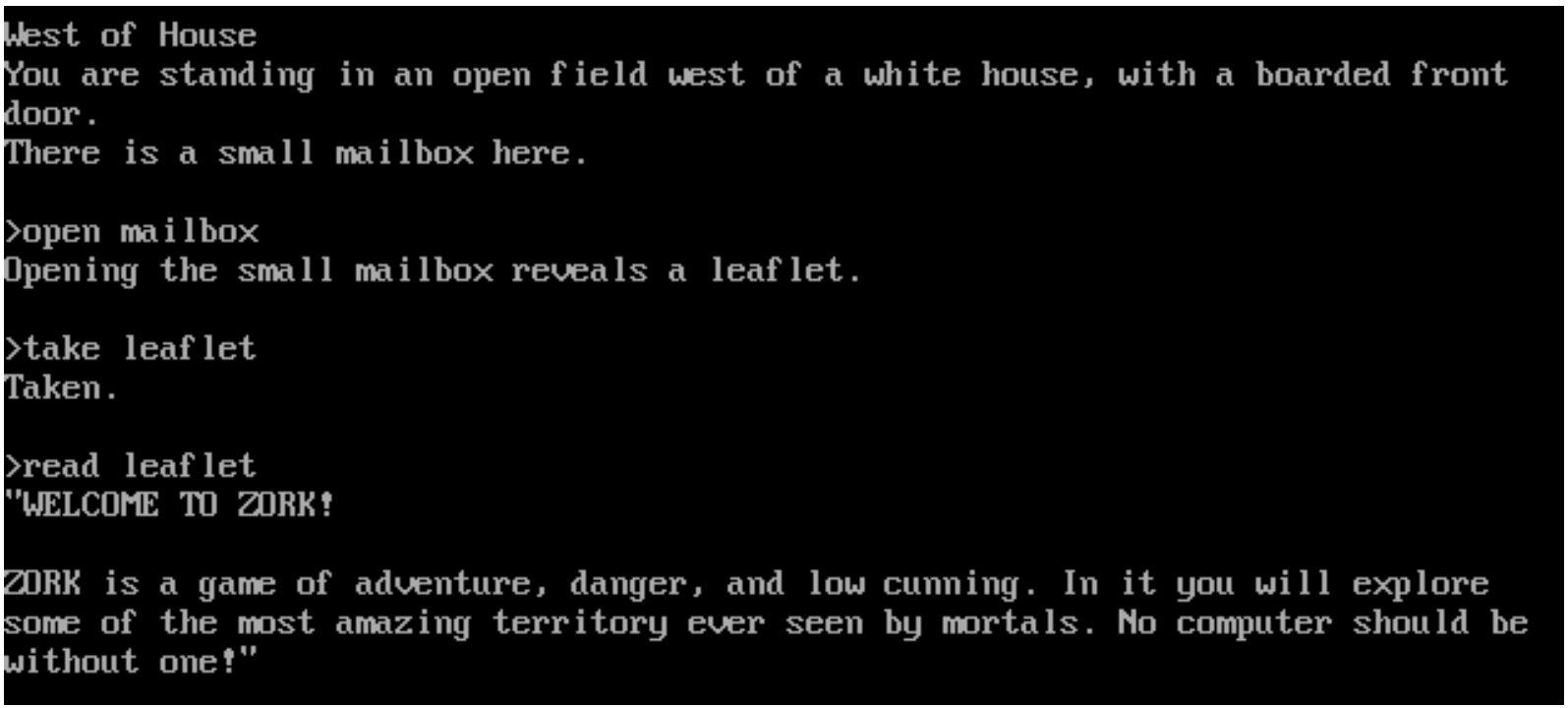
Affordances for Sword from Fulda et. al.

Our algorithm		Co-occurrence		Concept Net	
vanquish	impale	have	die	kill	harm
duel	battle	make	cut	parry	fence
unsheath	behead	kill	fight	strike	thrust
summon	wield	move	use	slash	injure
overpower	cloak	destroy	be	look cool	cut

`https://www.microsoft.com/en-us/research/project/textworld/`

Fulda, Nancy, Daniel Ricks, Ben Murdoch, and David Wingate. "What can you do with a rock? Affordance extraction via word embeddings." In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1039-1045. AAAI Press, 2017.

Sample game - Zork I



Beyond games

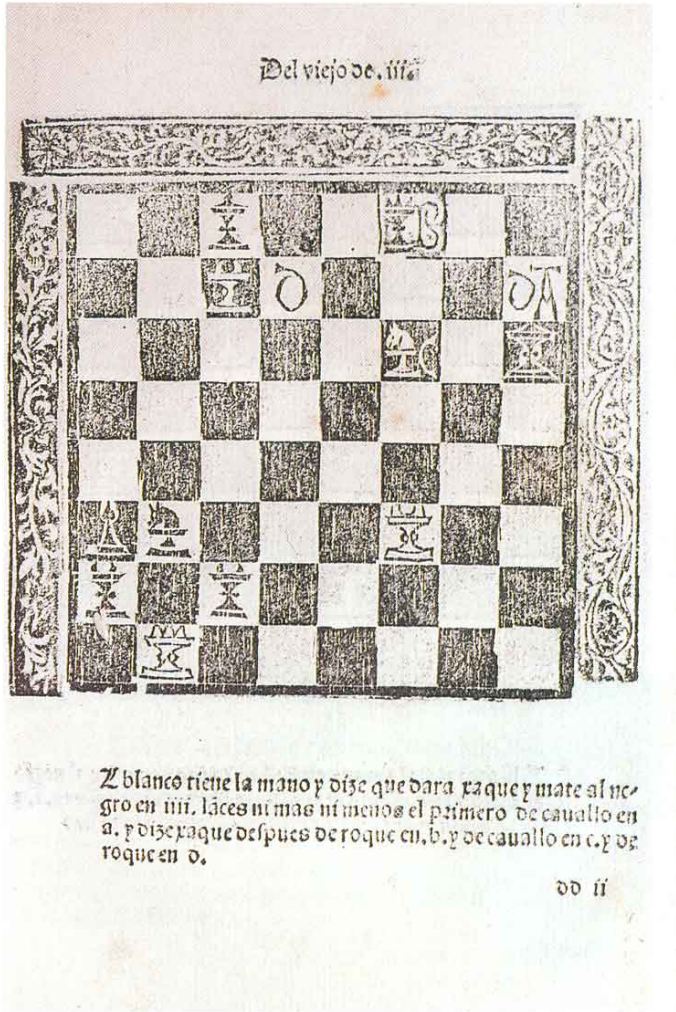
- Dialogue / Chatbots
- Try to get the user to agree to something
- (Mostly) assumes we have have a model of a user - what if the user is adapting back to the agent?

Li, Jiwei, et al. "Deep Reinforcement Learning for Dialogue Generation." Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

Young, Steve, et al. "POMDP-based statistical spoken dialog systems: A review." Proceedings of the IEEE 101.5 (2013): 1160-1179.

Transition probabilities

- Normal RL assumes they are fixed
  - *Augustinian Universe*
- But what if you are not alone in the world? What if there are other agents (that might not like you that much?)
  - *Manichean Universe*
- Or a more accurate description might be *Augustinian vs Valentinian vs Sethian*
- Game Theory...



Standard RL algorithms crush

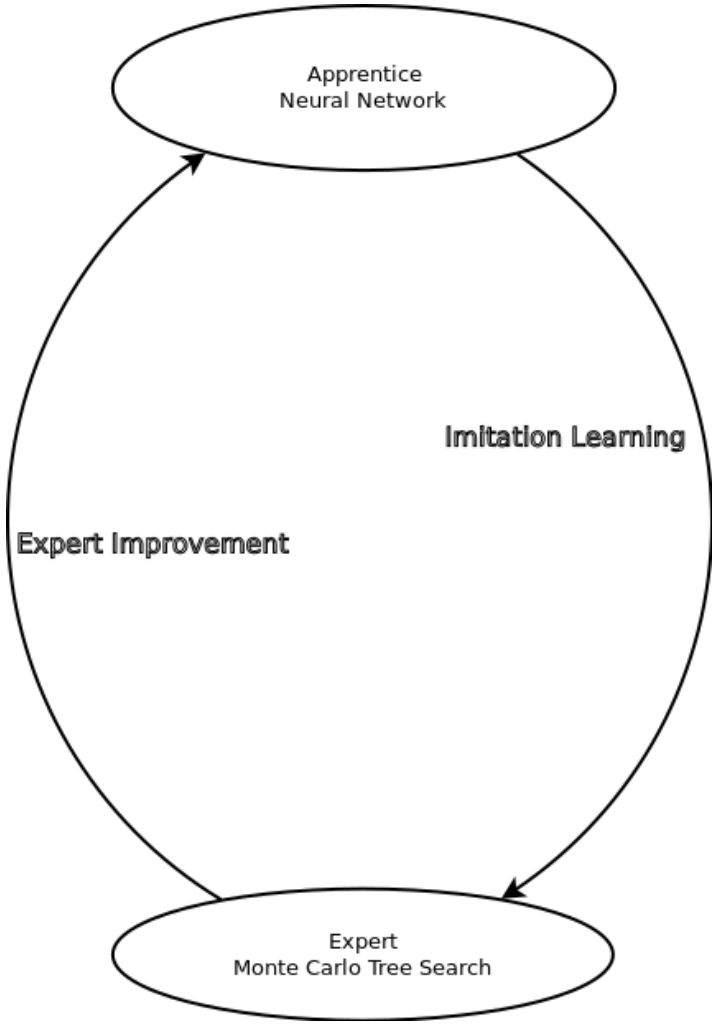
- Again, same problem with function approximation
  - But it's now more intense!
  - There is incentive to explore areas of the search space you have forgotten about
- Intense cycling of policies
- You can detect the amount of cycling your algorithm does, but it won't help you with solving the problem

---

Samothrakis, Spyridon, et al. "Coevolving game-playing agents: Measuring performance and intransitivities." IEEE Transactions on Evolutionary Computation 17.2 (2013): 213-226.

Expert Iteration

- Algorithm released almost a year ago by a UCL group
- At the same time, DeepMind released AlphaGo Zero
- You have a game player (the *apprentice* - possibly in the form of a neural network)
- You self-play games and you collect the states you visited
- You use MCTS as the *expert* to improve the policy
  - Requires a model



Anthony, Thomas, Zheng Tian, and David Barber. "Thinking fast and slow with deep learning and tree search." Advances in Neural Information Processing Systems. 2017.

Multiple agents and partial observability

- Separate streams of game theoretic algorithms
  - Fictitious play
  - Counterfactual Regret Minimization
- But seems like game theoretical and tree search/RL methods are converging
- Mixed Strategies, i.e. optimal policies not deterministic
  - "Bluffing"

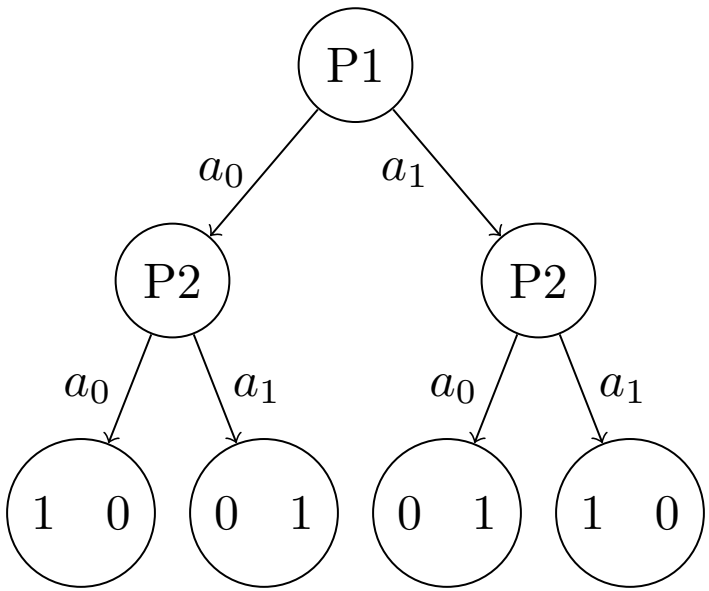
---

https://www.flickr.com/photos/valerianasolaris/3700647391/sizes/l/

Brown, Noam, Tuomas Sandholm, and Brandon Amos. "Depth-Limited Solving for Imperfect-Information Games." arXiv preprint arXiv:1805.08195 (2018).

Matching Pennies

- Both actions happen at the same time
- Same action, P1 wins
- Different actions, P2 wins
- Minmax equilibrium is to play half play
  $p(P1, a_0) = 0.5, p(P1, a_1) = 0.5$
- Wrecks havoc at Q-learning



Multiple agents break standard RL tricks

- Some of the old tricks are no longer viable
- e.g. the almost ubiquitous "experience replay" is harder to use
- Data in the archive because "stale", because the transition dynamics have changed
- But even if you do calculate Nash Equilibria
  - When you deploy your agent, your opponent almost always has a slight bias
  - You actually need all the non-dominated policies

Foerster, Jakob, et al. "Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning." International Conference on Machine Learning. 2017.

Speculation II - Multi-agent learning is problematic because...

- **The adversarial transitions create even more covariate shift and concept drift**
  - They tend to get you to places you'd rather forget
  - Multiple correct deterministic polices for the same state
- **Learn a subset of deterministic policies and adapt online**
- **Multiple function approximators for different parts of the state space**

What are rewards?

- Preference functions
- An agent prefers some states/actions combinations over other
- Can we train the reward function?
- Game link - rewards are like a wish spell!
  - "I wish to have all the money in the world"
  - You don't really mean it...

Externalities

- An agent's single minded pursue of rewards can cause negative unforeseen side-effects
- Humans are surprisingly good at not behaving like that
  - We criticise, change and patch our rewards
  - Gordon Gecco type of behaviours are rare
- The reward function might need to change as the agent learns more
  - There are limits to what can be achieved
- Rewards change through the life of the agent
- You should be able to re-use your already existing knowledge to adapt

Q/V/Policies - "What-if"

- If rewards change, your Q/V/Policies are almost useless
- Your forward and inverse models are not!
- You can now search the model before going to the real world
  - Planning as learning

Ladder of Causation

- Imagination (What if I had I had wings?)
  - What if I had not taken the aspirin? Why?
- Intervention (What if I do X?)
  - If I take an aspirin, will my headache be cured?
- Correlations (What if I see X?)
  - Neighbour took aspirin, headache was gone

Pearl, Judea, and Dana Mackenzie. The Book of Why: The New Science of Cause and Effect. Basic Books, 2018.

Concluding...

- Maybe to be able to act optimally we need to create a "causal language" in terms of abstractions
  - What-if/what-if not
  - Communicate with other mini-agents
  - Independent components
- We need lot's of small agents, possibly with conflicting rewards
- Sensory input needs to be combination of words, mechanics and lower level sensors
- Rewards change

Speculation III - RL does not scale because the world modeling we do is broken

- We erroneously treat the MDP as an abstraction of universal dynamics
  - Not explicitly, but implicitly
- A well defined action space is completely artificial and only found in certain games
  - Actions need to be discovered
- The whole universe needs to be described as an interplay between acting objects
  - Explained using causes of effects
  - What-if on everything!
- Models learned should transfer

Minsky, Marvin. Society of mind. Simon and Schuster, 1988.

Role Playing Games as the ultimate benchmark?

- Fantasy worlds with clear mechanics
  - Counterfactuals, the ability to think thoughts about things have never existed
- Combine language, visuals, player actions
- Massive dialogues with mechanics that will force your function approximator to create state
  - Or memory mechanisms

---

https://aeon.co/ideas/dungeons-and-dragons-not-chess-and-go-why-ai-needs-roleplay