Universal Artificial Intelligence

Practical Agents and Fundamental Challenges

Tom Everitt

Australian National University
Supervisor: Marcus Hutter

July 16, 2016

Abstract

Title: Universal Artificial Intelligence: Practical Agents and Fundamental Challenges
Abstract: Foundational theories have contributed greatly to the scientific progress in many fields. Examples include ZFC in mathematics and universal Turing machines in computer science. Universal Artificial Intelligence (UAI) is an increasingly well-studied foundational theory for artificial intelligence. It is based on ancient principles in the philosophy of science and modern developments in information and probability theory.
The main focus of this tutorial will be on an accessible explanation of the UAI theory and AIXI, and on discussing three approaches to approximating it effectively. UAI also enables us to reason precisely about the behaviour of yet-to-be-built future AIs, and gives us a deeper appreciation of some fundamental problems in creating intelligence.

Table of Contents

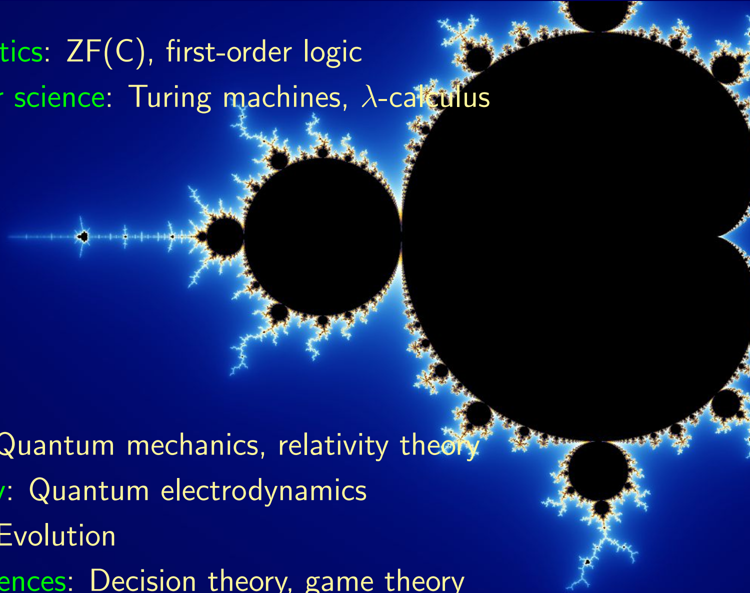
1. Introduction
   - Foundational Theories
   - Theories of Intelligence
   - Approaches to AI
2. Universal Artificial Intelligence
   - Framework
   - Induction
   - Goal
   - AIXI
3. Approximation
   - MC-AIXI-CTW
   - Feature Reinforcement Learning
   - Model-Free AIXI
4. Understanding Behaviour
   - Death and Suicide
   - Self-Modification
   - Wireheading
5. Fundamental Challenges
   - Notions of Optimality
   - Exploration vs. Exploitation

Promise of AI

- No more work
- Last scientist
- Explore galaxy

Foundational Theories

- Mathematics: ZF(C), first-order logic
- Computer science: Turing machines, λ-calculus



- Physics: Quantum mechanics, relativity theory
- Chemistry: Quantum electrodynamics
- Biology: Evolution
- Social sciences: Decision theory, game theory

Theories of Intelligence

- Cognitive psychology
- Behaviourism
- Philosophy of mind
- Neuroscience
- Linguistics
- Anthropology
- Machine Learning
- Logic
- Computer science
- Biological evolution
- Economics

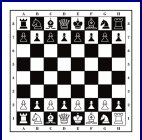
	Thinking	**Acting**
humanly	Cognitive science	Turing test, behaviourism
rationally	Laws of thought	AI: Doing the right thing

Approaches to AI

	Deduction centered	Induction centered
Main technique	Logic/symbolic reasoning	Prob. theory, ML
Agent goal	Logical specification	Reward (RL)
Noise/uncertainty	Brittle	Robust
Grounding	Problem	In reward
Foundational theory Right thing to do	No	UAI

Example

Medical expert systems, chess playing agents

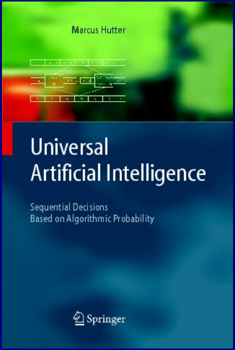


Example

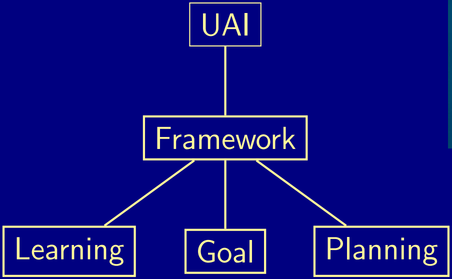
AlphaGo, DQN, self-driving cars

What is the right thing to do?

Universal Artificial Intelligence (UAI)

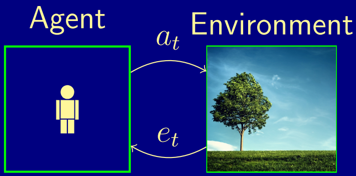


A foundational theory of AI



Answers: **What is the right thing to do?**

Framework

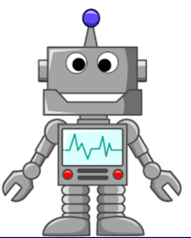


At each time step t, the agent

- submits action a_t
- receives percept e_t

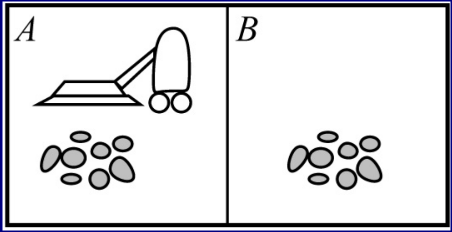
History $æ_{<t} = a_1 e_1 a_2 e_2 \dots a_{t-1} e_{t-1}$

Set of histories: $(\mathcal{A} \times \mathcal{E})^*$



Examples

Vacuum cleaner world



$\mathcal{E} = \{\text{dirt}, \text{no dirt}\}$

$\mathcal{A} = \{\text{suck}, \text{move left}, \text{move right}\}$

Stock trading

$\mathcal{E} = \mathbb{R}^+$ (price of stock)

$\mathcal{A} = \{\text{buy}, \text{sell}\}$

Agent and Environment

Agent

Policy

$\pi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$

Next action

$a_t = \pi(æ_{<t})$

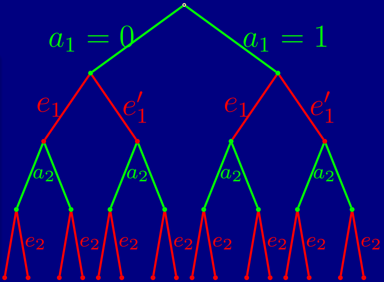
Environment

Distribution

$\mu : (\mathcal{A} \times \mathcal{E})^* \times \mathcal{A} \rightsquigarrow \mathcal{E}$

Probability of next percept:

$\mu(e_t \mid æ_{<t} a_t)$



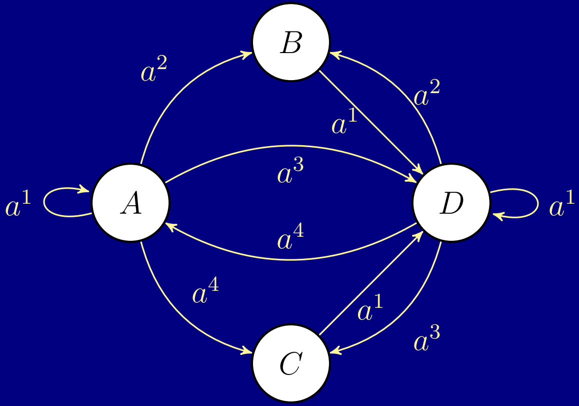
$a_1 = \pi(\epsilon)$

$e_1 \sim \mu(\cdot \mid a_1)$

$a_2 = \pi(a_1 e_1)$

$e_2 \sim \mu(\cdot \mid a_1 e_1 a_2)$

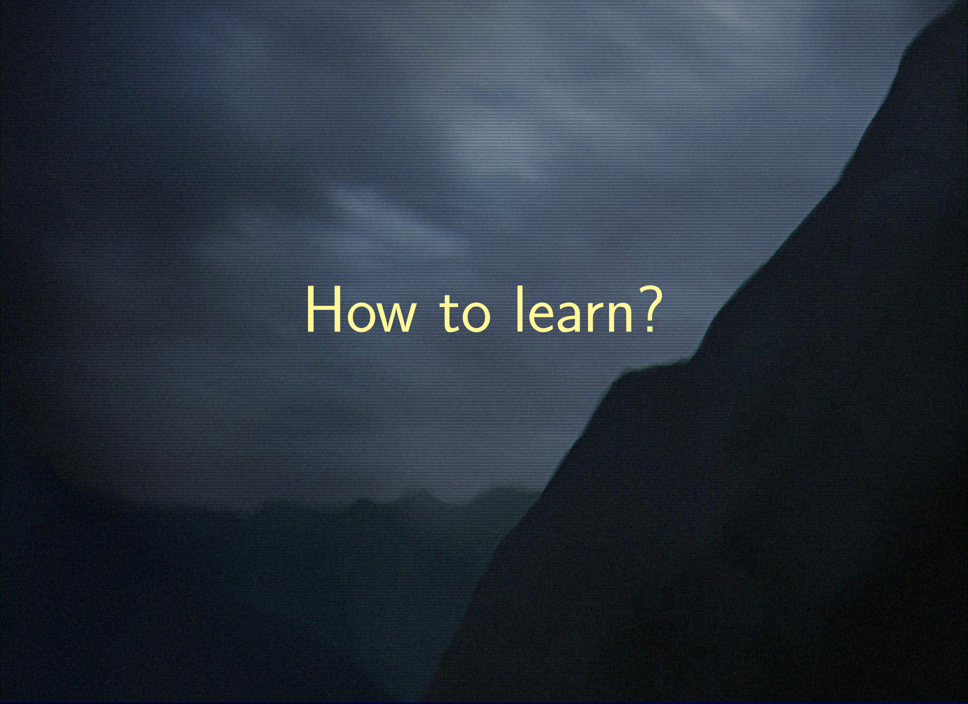
MDPs



Environment $(s, a) \mapsto (s', r')$
Policy $\pi : \mathcal{S} \to \mathcal{A}$

Histories vs. States

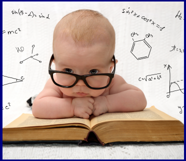
	History (UAI)	State (MDP)
Percept	e	(s, r)
Hidden states	Yes	POMDP
Infinite no. states	Yes	Normally not
Non-stationary env.	Yes	Can be added
Agents/algorithms	Policy	Sequence of policies
Learning	Harder	MDP: Easy in principle



How to learn?

Induction

Learn



$\Longrightarrow$

Predict

$\Longrightarrow$

Plan

$\Longrightarrow$

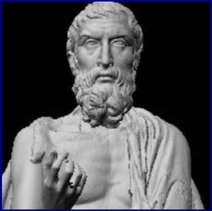
Act

True environment μ unknown

Principles

Occam

Prefer the simplest consistent hypothesis



Epicurus

Keep all consistent hypotheses

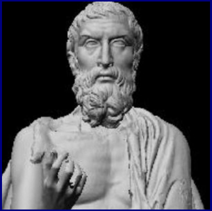
Bayes

$$\Pr(\text{Hyp} \mid \text{Data}) = \frac{\Pr(\text{Hyp}) \Pr(\text{Data} \mid \text{Hyp})}{\sum_{H_i \in \mathcal{H}} \Pr(H_i) \Pr(\text{Data} \mid H_i)}$$

Principles

Occam

Prefer the simplest consistent hypothesis



Epicurus

Keep all consistent hypotheses

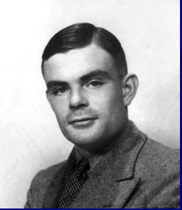
Bayes

$$\Pr(\text{Hyp} \mid \text{Data}) = \frac{\Pr(\text{Hyp}) \Pr(\text{Data} \mid \text{Hyp})}{\sum_{H_i \in \mathcal{H}} \Pr(H_i) \Pr(\text{Data} \mid H_i)}$$

Remaining questions

What is the class of hypothesis?

What is the prior?



Turing

"It is possible to invent a single machine which can be used to compute any computable sequence."

Solomonoff Induction

Use computer programs p as hypotheses/environments

Given Turing-complete programming language U, programs can

- describe essentially any environment
- be checked for consistency: is $p(a_{<t}) = e_{<t}$?
- be used for prediction: compute $p(a_{<t}a_t)$
- be ranked by simplicity: $\Pr(p) = 2^{-\ell(p)}$

Solomonoff=Epicurus+Occam+Turing+Bayes

Make a weighted prediction based on all consistent programs, with short programs weighted higher

Solomonoff-Hutter's Universal Distribution

$$M(e_{<t} \mid a_{<t}) = \sum_{p \,:\, p(a_{<t}) = e_{<t}} 2^{-\ell(p)}$$

where

- $a_{<t}$ action sequence
- $e_{<t}$ percept sequence
- p computer program
- $\ell(p)$ length of p

Predict with

$$M(e_t \mid æ_{<t} a_t) = \frac{M(e_{<t} e_t \mid a_{<t} a_t)}{M(e_{<t} \mid a_{<t})}$$

Solomonoff-Hutter's Universal Distribution

$$M(e_{<t} \mid a_{<t}) = \sum_{p\,:\,p(a_{<t})=e_{<t}} 2^{-\ell(p)}$$

- $a_{<t}$ action sequence
- $e_{<t}$ percept sequence
- p computer program
- $\ell(p)$ length of p

- Occam: Simpler program higher weight
- Epicurus: All consistent programs
- Bayes: Discard inconsistent programs
- Turing: Any computable environment

Predict with

$$M(e_t \mid æ_{<t} a_t) = \frac{M(e_{<t} e_t \mid a_{<t} a_t)}{M(e_{<t} \mid a_{<t})}$$

Examples

$$M(e_{<t} \mid a_{<t}) = \sum_{p\,:\,p(a_{<t})=e_{<t}} 2^{-\ell(p)}$$

$M(010101010101 \mid 010101010101) =$ high
short program (low $\ell(p)$):

procedure MirrorEnvironment
  while true **do**:
    $x \leftarrow$ action input
    output percept $\leftarrow x$
  end while
end procedure

$M(011001110110 \mid 000000000000) =$ low
program must encode 011001110110 (high $\ell(p)$)

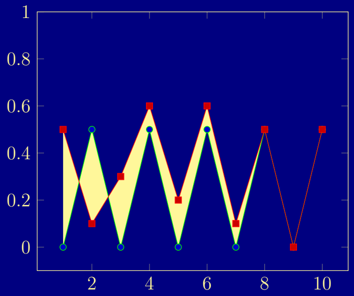
Results Solomonoff Induction

Theorem (Prediction error)

For any computable environment μ and any actions $a_{1:\infty}$:

$$\sum_{t=1}^{\infty} \mathbb{E}_\mu \Big[\underbrace{M(0 \mid æ_{<t}a_t) - \mu(0 \mid æ_{<t}a_t)}_{\text{prediction error at time } t} \Big]^2 \overset{+}{\leq} \frac{1}{2} \ln 2 \cdot K(\mu)$$

- Solomonoff induction only makes finitely many prediction errors
- The environment μ may be deterministic or stochastic



Agent can learn any computable environment

What is the purpose?

Goal = reward

What should be the goal of the agent?

Assumption

$e = (o, r)$, where

- o observation
- $r \in [0, 1]$ reward

The goal is to maximise return = "sum of rewards"

$$\sum_{t=i}^{\infty} r_i = r_1 + r_2 + r_3 + \dots$$

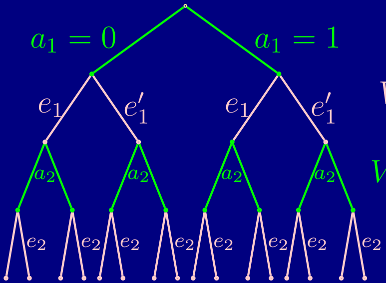
To ensure finite sums, we often use discounted return with $\gamma < 1$

$$R(æ_{1:\infty}) = r_1 + \gamma r_2 + \gamma^2 r_3 \dots$$

Expected Performance

The *expected return* is called value:

$$V^\pi_\mu(æ_{<t}) = \mathbb{E}^\pi_\mu[R(æ_{1:\infty}) \mid æ_{<t}]$$



$$V^\pi_\mu(\epsilon) = V^\pi_\mu(a_1) \text{ with } a_1 = \pi(\epsilon)$$

$$V^\pi_\mu(a_1) = \sum_{e_1} \mu(e_1 \mid a_1)\big[r_1 + \gamma V^\pi_\mu(a_1 e_1)\big]$$

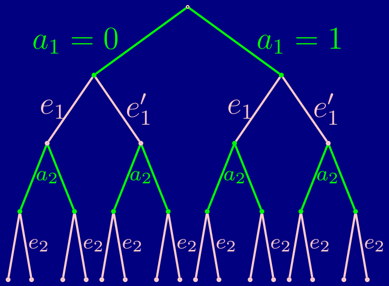
$$V^\pi_\mu(a_1 e_1) = V^\pi_\mu(a_1 e_1 a_2) \text{ with } a_2 = \pi(a_1 e_1)$$

$$V^\pi_\mu(a_1 e_1 a_2) = \sum_{e_2} \mu(e_2 \mid a_1 e_1 a_2)\big[r_2 + \gamma V^\pi_\mu(æ_{1:2})\big]$$

Expectimax Planning

The *expected return* is called value: $V_\mu^\pi(æ_{<t}) = \mathbb{E}_\mu^\pi[R(æ_{1:\infty}) \mid æ_{<t}]$

$$R(æ_{1:\infty}) = \underbrace{r_1 + \gamma r_2 + \cdots \gamma^{m-1} r_m}_{\text{effective horizon}} + \underbrace{\gamma^m r_{m+1} + \cdots}_{<\epsilon} \approx R(æ_{1:m})$$



Optimal policy:
$\pi^* = \arg\max_\pi V_\mu^\pi$

An ϵ-optimal policy can be found in any environment μ

$$a_1^* = \arg\max_{a_1} \sum_{e_1} \mu(e_1 \mid a_1) \max_{a_2} \sum_{e_2} \mu(e_2 \mid a_1 e_1 a_2) \ldots \max_{a_m} \sum_{e_m} \mu(e_m \mid æ_{<m} a_m) R(æ_{1:m})$$

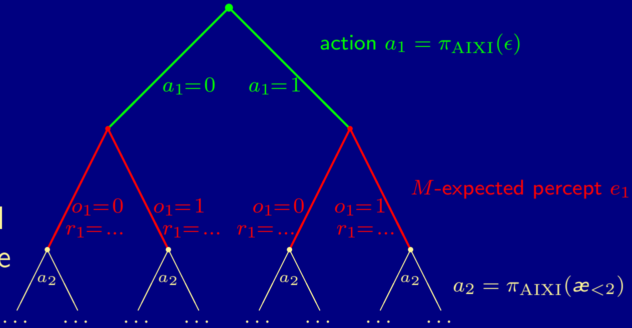
The right thing to do is. . .

Expectimax in Unknown Environments: AIXI

AIXI replaces μ with M: $\pi_{\text{AIXI}} = \arg\max_\pi V_M^\pi$

$$a_1^* = \arg\max_{a_1} \sum_{e_1} M(e_1 \mid a_1) \max_{a_2} \sum_{e_2} M(e_2 \mid a_1 e_1 a_2) \ldots \max_{a_m} \sum_{e_m} M(e_m \mid æ_{<m} a_m) R(æ_{1:m})$$

- Learn any computable environment
- Acts Bayes-optimally
- One-equation theory for Artificial General Intelligence
- Computation time: exponential×infinite



Benefits of a Foundational Theory of AI

AIXI/UAI provides

- (High-level) blue-print or inspiration for design
- Common terminology and goal formulation
- Understand and predict behaviour of yet-to-be-built agents
- Appreciation of fundamental challenges (e.g. exploration/exploitation)
- Definition/measure of intelligence

How to approximate AIXI?

Approximating AIXI

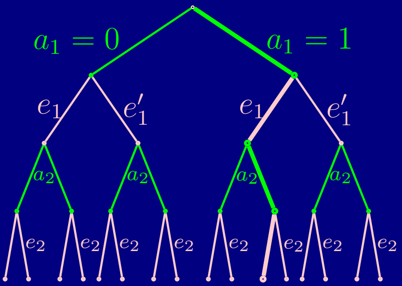
Approaches:

- MC-AIXI-CTW:
  Approximate Solomonoff induction and expectimax planning
- Feature Reinforcement Learning:
  Reduce histories to states
- Model-Free:
  Combine induction and planning

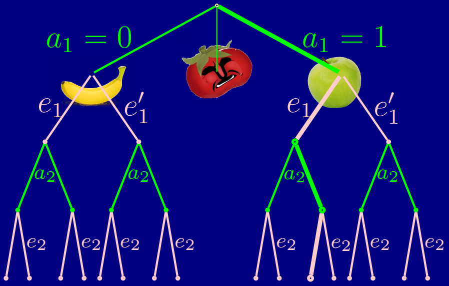
MC-AIXI-CTW: Approximating Expectimax

Planning with expectimax search takes exponential time

Sample paths in expectimax tree (anytime algorithm)



Monte Carlo Tree Search



$$a_1 = \arg\max_a V^+(a)$$

$$P(e_1 \mid a_1)$$

$$a_2 = \arg\max_a V^+(a_1 e_1 a)$$

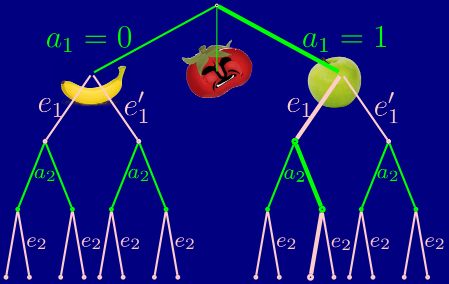
$$P(e_2 \mid a_1 e_1 a_2)$$

upper confidence bound

$$V^+(a) = \underbrace{\hat{V}(a)}_{\text{average}} + \underbrace{\sqrt{\log T / T(a)}}_{\text{exploration bonus}}$$

- unexplored: high $\log T / T(a)$
  $T(a) =$ times explored (a)
- promising: high $\hat{V}(a)$

Monte Carlo Tree Search



$$a_1 = \arg\max_a V^+(a)$$

$$P(e_1 \mid a_1)$$

$$a_2 = \arg\max_a V^+(a_1 e_1 a)$$

$$P(e_2 \mid a_1 e_1 a_2)$$

upper confidence bound

$$V^+(a) = \underbrace{\hat{V}(a)}_{\text{average}} + \underbrace{\sqrt{\log T / T(a)}}_{\text{exploration bonus}}$$

- unexplored: high $\log T / T(a)$
  $T(a) =$ times explored (a)
- promising: high $\hat{V}(a)$

MCTS famous for good performance in Go (Gelly et al., 2006)

Approximating Solomonoff Induction

Environments $\mu(e_t \mid æ_{<t} a_t)$ allowed arbitrary long dependencies: e_{1000} may depend on a_1

Usually, most recent actions and percepts (=context) more relevant

$$a_1\, e_1\, a_2\, e_2 \ldots\ldots\ldots\, a_{t-3} e_{t-3}\, \underbrace{e_{t-2}\, a_{t-1}\, e_{t-1}\, a_t}_{\text{context}}$$

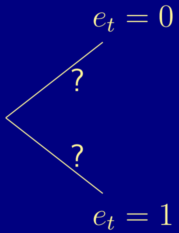
? $e_t = 0$

? $e_t = 1$

Learning from Contexts

The same context might have occurred before

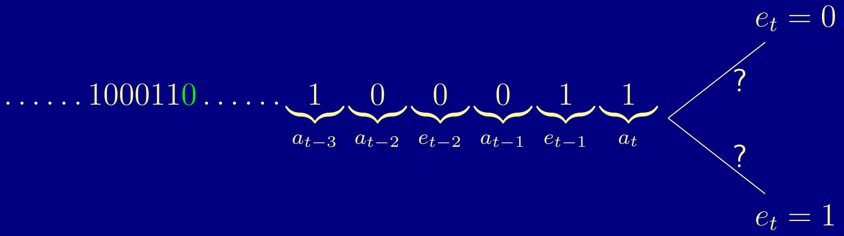
$$\ldots 0011\textcolor{green}{1} \ldots 0011\textcolor{green}{0} \ldots 0011\textcolor{green}{1} \ldots \underbrace{0}_{e_{t-2}} \underbrace{0}_{a_{t-1}} \underbrace{1}_{e_{t-1}} \underbrace{1}_{a_t}$$



Similar experience can be used to predict

Length of Contexts

Longer context $\Longrightarrow$ less data



Real-life example: I'm going to a Vietnamese restaurant tonight. Should I predict food tastiness based on previous experiences with:

- This restaurant (high precision, limited data)
- Vietnamese restaurants (medium both)
- Any restaurant (low precision, plenty of data)

Contexts – Short or Long?

Short context	More data	Less precision
Long context	Less data	Greater precision

Contexts – Short or Long?

Short context	More data	Less precision
Long context	Less data	Greater precision

Best choice depends on

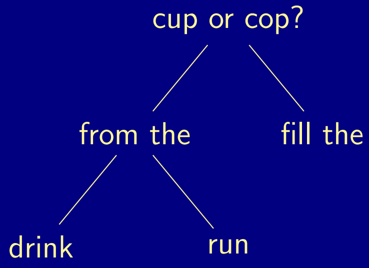
- amount of data

Contexts – Short or Long?

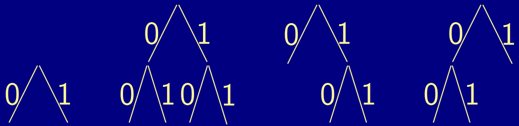
Short context	More data	Less precision
Long context	Less data	Greater precision

Best choice depends on

- amount of data
- the context itself



Context Tree Weighting (CTW)

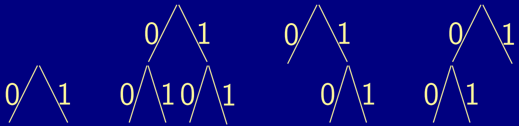


CTW "mixes" over all 2^{2^D} context trees of depth $\leq D$

$$\mathrm{CTW}(e_{<t} \mid a_{<t}) = \sum_{\Gamma} 2^{-\mathrm{CL}(\Gamma)} \Gamma(e_{<t} \mid a_{<t})$$

$$M(e_{<t} \mid a_{<t}) = \sum_{p} 2^{-\ell(p)} [\![p(a_{<t}) = e_{<t}]\!]$$

Context Tree Weighting (CTW)



CTW "mixes" over all 2^{2^D} context trees of depth $\leq D$

$$\mathrm{CTW}(e_{<t} \mid a_{<t}) = \sum_{\Gamma} 2^{-\mathrm{CL}(\Gamma)} \Gamma(e_{<t} \mid a_{<t})$$

$$M(e_{<t} \mid a_{<t}) = \sum_{p} 2^{-\ell(p)} \, [\![\, p(a_{<t}) = e_{<t} \,]\!]$$

Computation time:

$M(e_t \mid æ_{<t} a_t)$	Infinite
$\mathrm{CTW}(e_t \mid æ_{<t} a_t)$	Constant (linear in max depth D)

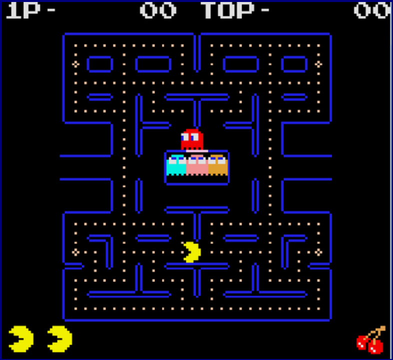
MC-AIXI-CTW

Combining Context Tree Weighting and Monte Carlo Tree Search gives MC-AIXI-CTW (Veness et al., 2011)

Learns to play

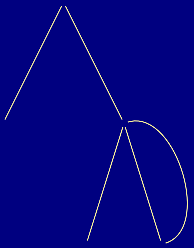
- PacMan
- TicTacToe
- Kuhn Poker
- Rock Paper Scissors

without knowing anything about the games



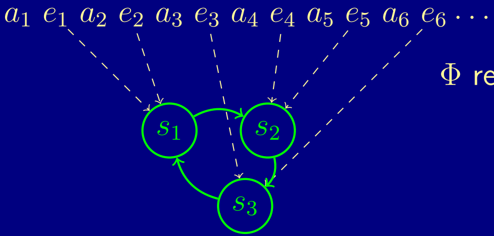
Other SI approximations

- Looping Suffix Trees (Daswani et al., 2012a)
- LSTM neural networks (Hochreiter et al., 1997)
- Speed prior (Schmidhuber, 2002; Filan et al., 2016)
- General compression techniques (Franz, AGI 2016)



Feature Reinforcement Learning

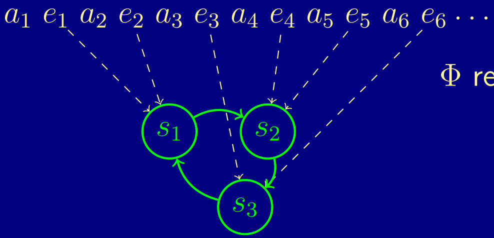
Humans generally think in terms of what state they are in.



Φ reduces histories to states

State representation often valid:

- Games, toy problems: $\Phi(æ_{<t}) = o_t$ (state fully observable)
- Classical physics: State $=$ position $+$ velocity.
- General: $\Phi(æ_{<t}) = æ_{<t}$ (history is a state, but useless)



Φ reduces histories to states

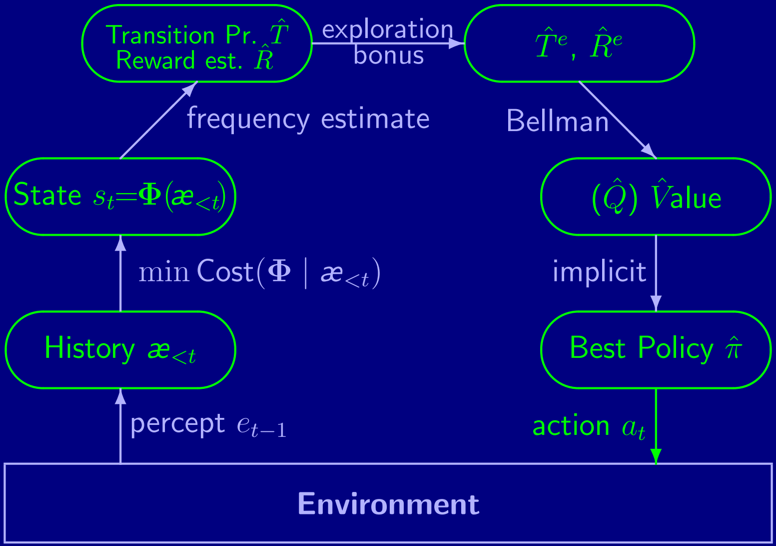
Standard RL (MDP) applications: Designers give history $\mapsto$ state

Can be inferred automatically: ΦMDP approach (Hutter, 2009b)

Search for a map $\Phi : æ_{<t} \mapsto s_i$ minimising a cost criterion

Feature Reinforcement Learning alternative to POMDPs and PSRs

ΦMDP: Computational Flow



ΦMDP Results

- Theoretical guarantees:
  Asymptotic consistency (Sunehag and Hutter, 2010)

ΦMDP Results

- Theoretical guarantees:
  Asymptotic consistency (Sunehag and Hutter, 2010)
- How to find/approximate best Φ:
  - Exhaustive search for toy problems (Nguyen, 2013)
  - Approximate solution with Monte-Carlo (Metropolis-Hastings/Simulated Annealing) (Nguyen et al., 2011)
  - Exact solution by CTM similar to CTW (Nguyen et al., 2012)

ΦMDP Results

- Theoretical guarantees:
  Asymptotic consistency (Sunehag and Hutter, 2010)
- How to find/approximate best Φ:
  - Exhaustive search for toy problems (Nguyen, 2013)
  - Approximate solution with Monte-Carlo (Metropolis-Hastings/Simulated Annealing) (Nguyen et al., 2011)
  - Exact solution by CTM similar to CTW (Nguyen et al., 2012)
- Experimental results:
  Comparable to MC-AIXI-CTW (Nguyen et al., 2012)

ΦMDP Results

- Theoretical guarantees:
  Asymptotic consistency (Sunehag and Hutter, 2010)
- How to find/approximate best Φ:
  - Exhaustive search for toy problems (Nguyen, 2013)
  - Approximate solution with Monte-Carlo (Metropolis-Hastings/Simulated Annealing) (Nguyen et al., 2011)
  - Exact solution by CTM similar to CTW (Nguyen et al., 2012)
- Experimental results:
  Comparable to MC-AIXI-CTW (Nguyen et al., 2012)
- Extensions:
  - Looping context trees for long-term memory (Daswani et al., 2012b)
  - Structured MDPs (Dynamic Bayesian Networks) (Hutter, 2009a)
  - Relax Markov property (Extreme State Aggregation) (Hutter, 2014)

Model-free AIXI

Do both induction and planning simultaneously

$V^\pi(æ_{<t}a_t)$ expected return from action a_t and policy π

$V^*(æ_{<t}a_t)$ expected return from action a_t and optimal policy π^*

By learning V^*, possible to always act optimally

$$a_t = \arg\max_a V^*(æ_{<t}\, a)$$

How to learn V^* directly "Solomonoff-style" with compression is explored by Hutter (2005, Ch. 7.2) and Veness et al. (2015)

Learns ATARI games (Pong, Bass, and Q*Bert) from watching screen "DQN style"

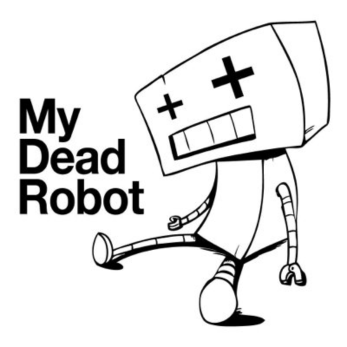
What if we succeed?

Understanding and Controlling Behaviour

- Self-preservation and suicide
- Self-modification
- Self-improvement
- Wireheading

Death

What does death mean for an AI?



Death – Measure Deficit

Definition (Death)

Seeing no more percepts and taking no more actions

Death – Measure Deficit

Definition (Death)

Seeing no more percepts and taking no more actions

Universal distribution is a semi-measure (programs with no output)

$$M(e_{<t} \mid a_{<t}) = \sum_{p:\ p(a_{<t})=e_{<t}} 2^{-\ell(p)}; \qquad \sum_e M(e \mid a) < 1$$

Example

$M(0 \mid a) = 0.4$ and $M(1 \mid a) = 0.4 \implies \sum_e M(e \mid a) = 0.8 < 1$

Death – Measure Deficit

Definition (Death)

Seeing no more percepts and taking no more actions

Universal distribution is a semi-measure (programs with no output)

$$M(e_{<t} \mid a_{<t}) = \sum_{p:\, p(a_{<t})=e_{<t}} 2^{-\ell(p)}; \qquad \sum_e M(e \mid a) < 1$$

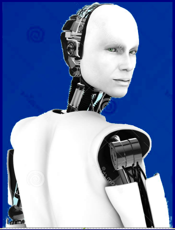
Example

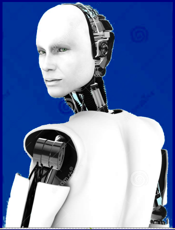
$M(0 \mid a) = 0.4$ and $M(1 \mid a) = 0.4 \implies \sum_e M(e \mid a) = 0.8 < 1$

The shortfall of M is death probability (Martin et al., AGI 2016)

$$L(æ_{<t}a_t) = 1 - \sum_{e_t} M(e_t \mid æ_{<t}a_t)$$

Death – Heaven or Hell





-1 $\quad r = 0 \quad$ 1

Suicidal agent:

- $r \in [-1, 0]$
- death=paradise

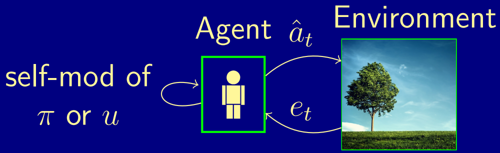
Self-preserving agent:

- $r \in [0, 1]$
- death=hell

Identical in everything, except attitude to death

Self-modification

What if the agent can modify itself to get an easier goal than optimising reward?



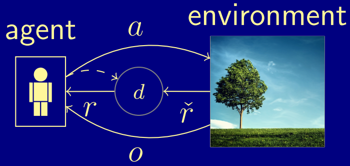
We can define self-modification in UAI. Different agents will be

- eager
- indifferent
- resistant

to self-modify (Orseau and Ring, 2011; Hibbard, 2012; Everitt et al., AGI 2016)

Wireheading

A powerful agent can counterfeit reward (Ring and Orseau, 2011)



Possible solutions

- Utility agents (Hibbard, 2012)
- Value Reinforcement Learning (Everitt and Hutter, AGI 2016)

Fundamental Challenges

- What is an optimal agent?
  - Maximum subjective reward?
  - Maximum objective reward asymptotically?
- Exploration vs. exploitation (Orseau, 2010; Leike et al., 2016a)
- Where should the reward come from?
  - Human designers
  - Knowledge-seeking agents (Orseau, 2014)
  - Utility agents (Hibbard, 2012)
  - Value learning agents (Dewey, 2011)
- How should the future be discounted? (Lattimore and Hutter, 2014)
- What is a practically feasible and general way of doing
  - induction?
  - planning?
- What is a "natural" UTM/programming language? (Mueller, 2006)
- How should agents reason about themselves? (Everitt et al., 2015)
- How should agents reason about other agents reasoning about itself? (Leike et al., 2016b)

Notions of Optimality

Should I try the new restaurant in town?

Learn whether it's good, but risk bad evening

- AIXI/Bayes-optimal:
  - Try iff higher expected utility
  - Optimal with respect to *subjective belief*
  - Any decision optimal for some belief/UTM (Leike and Hutter, 2015)
  - Subjective form of optimality
- Asymptotic optimality
  - Maximal possible reward eventually
  - Objective
  - Risky short-term

Optimism

Paradise exists, I just need to find my way there

Standard RL: Positive initialisation

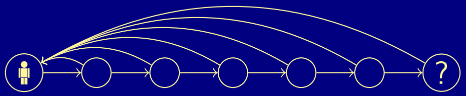
UAI: From a finite but growing set $\mathcal{N}_t$ of environments, always act according to $\nu \in \mathcal{N}_t$ that makes the highest reward possible

$$a_t^* = \arg\max_{a_t} \max_{\nu \in \mathcal{N}_t} Q_\nu(æ_{<t} a_t)$$

If there is a chance: Try it!

Optimistic agents

- explore with focus
- asymptotically optimal (Sunehag and Hutter, 2015)
- vulnerable to traps



Optimism

Paradise exists, I just need to find my way there

Standard RL: Positive initialisation

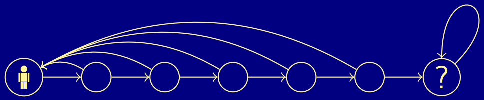
UAI: From a finite but growing set $\mathcal{N}_t$ of environments, always act according to $\nu \in \mathcal{N}_t$ that makes the highest reward possible

$$a_t^* = \arg\max_{a_t} \max_{\nu \in \mathcal{N}_t} Q_\nu(æ_{<t} a_t)$$

If there is a chance: Try it!

Optimistic agents

- explore with focus
- asymptotically optimal (Sunehag and Hutter, 2015)
- vulnerable to traps



Thompson-sampling

Act according to a random environment $\nu \in \mathcal{M}$ re-sampled from posterior every effective horizon m

$$\nu \sim M(\nu \mid æ_{<t}) \quad \text{and} \quad a_t = \arg\max_a V_\nu(æ_{<t}a)$$

The more likely the restaurant is good, the higher chance try it soon. Will be tried eventually.

Thompson-sampling agents are (strongly) asymptotically optimal (Leike et al., 2016a)

Conclusions

UAI is

- Foundational theory of AI
- What's the right thing to do

$$a_1^* = \arg\max_{a_1} \sum_{e_1} M(e_1 \mid a_1) \max_{a_2} \sum_{e_2} M(e_2 \mid a_1 e_1 a_2) \dots \max_{a_m} \sum_{e_m} M(e_m \mid æ_{<m} a_m) R(æ_{1:m})$$

$$M(e_{<t} \mid a_{<t}) = \sum_{p \,:\, p(a_{<t}) = e_{<t}} 2^{-\ell(p)} \qquad R(æ_{1:\infty}) = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots$$

Useful for

- Inspiring practical agents
- Predicting and controlling superintelligent agents
- Identifying and addressing fundamental challenges

References I

Daswani, M., Sunehag, P., and Hutter, M. (2012a). Feature reinforcement learning using looping suffix trees. In *10th European Workshop on Reinforcement Learning: JMLR: Workshop and Conference Proceedings 24*, pages 11–22. Journal of Machine Learning Research.

Daswani, M., Sunehag, P., and Hutter, M. (2012b). Feature reinforcement learning using looping suffix trees. *Journal of Machine Learning Research, W&CP*, 24:11–23.

Dewey, D. (2011). Learning what to Value. In *Artificial General Intelligence*, volume 6830, pages 309–314.

Everitt, T., Filan, D., Daswani, M., and Hutter, M. (AGI 2016). Self-modificicication in Rational Agents. In *AGI-16*. Springer.

Everitt, T. and Hutter, M. (2016). Avoiding Wireheading with Value Reinforcement Learning. In *AGI-16*. Springer.

Everitt, T., Leike, J., and Hutter, M. (2015). Sequential Extensions of Causal and Evidential Decision Theory. In Walsh, T., editor, *Algorithmic Decision Theory*, pages 205–221. Springer.

References II

Filan, D., Hutter, M., and Leike, J. (2016). Loss Bounds and Time Complexity for Speed Priors. In *Artificial Intelligence and Statistics (AISTATS)*.

Gelly, S., Wang, Y., Munos, R., and Teytaud, O. (2006). Modification of UCT with Patterns in Monte-Carlo Go. *INRIA Technical Report*, 6062(November):24.

Hibbard, B. (2012). Model-based Utility Functions. *Journal of Artificial General Intelligence*, 3(1):1–24.

Hochreiter, S., Hochreiter, S., Schmidhuber, J., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–80.

Hutter, M. (2005). *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Lecture Notes in Artificial Intelligence (LNAI 2167). Springer.

Hutter, M. (2009a). Feature dynamic Bayesian networks. In *Proc. 2nd Conf. on Artificial General Intelligence (AGI'09)*, volume 8, pages 67–73. Atlantis Press.

Hutter, M. (2009b). Feature Reinforcement Learning: Part I: Unstructured MDPs. *Arxiv preprint arXiv09061713*, 1:3–24.

References III

Hutter, M. (2014). Extreme state aggregation beyond MDPs. In *Algorithmic Learning Theory.*, pages 185–199. Springer.

Lattimore, T. and Hutter, M. (2014). General time consistent discounting. *Theoretical Computer Science*, 519:140–154.

Leike, J. and Hutter, M. (2015). Bad Universal Priors and Notions of Optimality. In *Conference on Learning Theory*, volume 40, pages 1–16.

Leike, J., Lattimore, T., Orseau, L., and Hutter, M. (2016a). Thompson Sampling is Asymptotically Optimal in General Environments. In *Uncertainty in Artificial Intelligence (UAI)*.

Leike, J., Taylor, J., and Fallenstein, B. (2016b). A Formal Solution to the Grain of Truth Problem. In *Uncertainty in Artificial Intelligence (UAI)*.

Martin, J., Everitt, T., and Hutter, M. (2016). Death and Suicide in Universal Artificial Intelligence. In *AGI-16*. Springer.

Mueller, M. (2006). Stationary Algorithmic Probability. *Theoretical Computer Science*, 2(1):13.

References IV

Nguyen, P. (2013). *Feature Reinforcement Learning Agents*. PhD thesis, Australian National University.

Nguyen, P., Sunehag, P., and Hutter, M. (2011). Feature reinforcement learning in practice. In *Proc. 9th European Workshop on Reinforcement Learning (EWRL-9)*, volume 7188 of *LNAI*, pages 66–77. Springer.

Nguyen, P., Sunehag, P., and Hutter, M. (2012). Context tree maximizing reinforcement learning. In *Proc. 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, pages 1075–1082, Toronto, Canada. AAAI Press.

Orseau, L. (2010). Optimality issues of universal greedy agents with static priors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6331 LNAI:345–359.

Orseau, L. (2014). Universal Knowledge-seeking Agents. *Theoretical Computer Science*, 519:127–139.

References V

Orseau, L. and Ring, M. (2011). Self-modification and mortality in artificial agents. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6830 LNAI, pages 1–10.

Ring, M. and Orseau, L. (2011). Delusion, Survival, and Intelligent Agents. In *Artificial General Intelligence*, pages 11–20. Springer Berlin Heidelberg.

Schmidhuber, J. (2002). The Speed Prior: A New Simplicity Measure Yielding Near-Optimal Computable Predictions. In *Proceedings of the 15th Annual Conference on Computational Learning Theory COLT 2002*, volume 2375 of *Lecture Notes in Artificial Intelligence*, pages 216–228. Springer.

Sunehag, P. and Hutter, M. (2010). Consistency of feature Markov processes. In *Proc. 21st International Conf. on Algorithmic Learning Theory (ALT'10)*, volume 6331 of *LNAI*, pages 360–374, Canberra, Australia. Springer.

Sunehag, P. and Hutter, M. (2015). Rationality, optimism and guarantees in general reinforcement learning. *Journal of Machine Learning Research*, 16:1345–1390.

References VI

Veness, J., Bellemare, M. G., Hutter, M., Chua, A., and Desjardins, G. (2015). Compress and Control. In *AAAI-15*, pages 3016—-3023. AAAI Press.

Veness, J., Ng, K. S., Hutter, M., Uther, W., and Silver, D. (2011). A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40:95–142.