Universal Artificial Intelligence

Department of Philosophy
Central South University
xieshenlixi@163.com

November 3, 2019

Philosophy

I'm a philosopher.
What do you expect from me?

Philosophy

Best philosophy in my eyes

- Leibniz — *What are the extent and limits of reason?* — Universal Characteristic & Rational Calculus.
- Cantor — *What is 'infinity'?* — Diagonal Method.
- Hilbert — *completeness/conservation/consistency/decidability.* — Gödel Incompleteness Theorems/Meta-Math.
- Turing — *What is 'effective procedure'?* — Computability.
- Kolmogorov — *What is 'simplicity'/'randomness'?* — AIT.
- Solomonoff — *What is learnable? How to learn?* — Universal Induction.
- Hutter — *What is 'intelligence'?* — UAI.

Philosophy

Could a tadpole hope that it will survive to be a frog?

- Metaphysics: What kinds of things exist? (e.g. tadpole, material things, mental states, relationships)
- Epistemology: What can we know and how do we know it? Can we ever know about the contents of "other" minds?
- Philosophy of mind: What are mental states and processes? Are certain material states sufficient to produce mental states?
- Ethics: How should we think about the rights of a tadpole? Do we have the right to cause them pain?
- Philosophy of science: What are scientific theories? Explanations? Evidence? Can theories ever be proved, or refuted, and if so how? What's the relationship between the development of new concepts and the development of new theories?
- Conceptual analysis: What do we mean by X? What does it mean to say that a tadpole hope hopes for something?
- ...

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Questions

- What is (artificial) intelligence?
- What does an intelligent system look like?
- Can there be a behavioural criterion for intelligence?
- Can computers think?
- Can connectionist networks think?
- Can physical symbol systems think?
- Do computing systems have "emergent" properties?
- Isn't changing the weight on a neural link a sort of symbol manipulation?
- Do computers have to be conscious to think?
- Are thinking computers mathematically possible?
- How will we know if we've done it?
- If we can do it, should we?

What is (Artificial) Intelligence?

What is AI?	**humanly**	**rationally**
Think	Cognitive Science	Laws of Thought
Act	Turing test, Behaviorism	Doing the Right Thing

Think like a human

- **Cognitive science:** Models of the human thinking processes.
- **Advantages:** Models of the human thinking processes./Intelligible.
- **Difficulties:** The best artificial design for an intelligent system need not mirror the human mind.

Physical symbol system hypothesis(Newell & Simon): a physical symbol system has the necessary and sufficient means for general intelligent action.
Any system (human or machine) exhibiting intelligence must operate by manipulating data structures composed of symbols.

Act like a human: Turing Test

- Alan M. Turing, "Computing Machinery and Intelligence"
- John R. Searle, "Minds, Brains, and Programs"

- Interrogator in one room, human in another, system in a third.
- Interrogator in one room, human in another, system in a third.
- Interrogator tries to guess which is which.
- Interrogator tries to guess which is which.
- Chinese Room argument. — Syntax is insufficient for dealing with semantics.

Needs: natural language, knowledge representation, automated reasoning, machine learning.
Difficulties: Ambiguous./Not constructive./Cannot be formalized mathematically.

If we don't use the Turing test, what measure should we use?

Think rationally: Logicist AI

- **Logic:** Automatic reasoning procedure.
- **Advantages:** Precise./Search algorithm.
- **Difficulties:** Formalization of informal knowledge./Computational Cost.

Act rationally: Agents

- **Rational agent:** Autonomous system, capable of perceiving and interacting with its environment, of exploration (information gathering), learning and adaptation, of formulating goals and designing plans to reach those goals. The agent is rational, in the sense that it acts to achieve the best (expected) outcome, according to a performance measure, conditioned to its knowledge of the world and given computational resources.

How do computers discover new knowledge?

- Fill in gaps in existing knowledge
- Emulate the brain
- Simulate evolution
- Systematically reduce uncertainty
- Notice similarities between old and new

Machine Learning

- Supervised Learning
  - Learn the relationship between "input" x and "output" y: search for a function f, such that $y \approx f(x)$
  - There is training data with labels available
    Regression: y is metric variable (with values in $\mathbb{R}$)
    Classification: y is categorical variable (unordered, discrete).
  - Semi-supervised learning: also uses available unlabeled data, e.g. assumes that similar inputs have similar outputs.
- Unsupervised Learning
  - There exist no outputs, search for patterns within the inputs x
    Clustering: find groups of similar items
    Dimensionality reduction: describe data in fewer features
    Outlier detection: what is out of the ordinary?
    Association rules: which things often happen together?
- Reinforcement learning

The Five Tribes of Machine Learning

Tribe	**Origins**	**Master Algorithm**
Symbolists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

Symbolists

- All intelligence can be reduced to manipulating symbols
- Logic, Decision trees
- Inverse deduction
- Easy to add knowledge
- Can combine knowledge, data, to fill in gaps
- Impossible to code everything in rules

Representation	Rules, trees, first order logic rules
Evaluation	Accuracy, information gain
Optimization	Top-down induction, inverse deduction
Algorithms	Decision trees, Logic programs

Connectionists

- Learning is what the brain does: reverse-engineer it
- Hebbian learning: Neurons that fire together, wire together
- Neural networks
- Backpropagation
- Can handle raw, high-dimensional data, constructs it own features
- Hard to add reasoning/explanations

Representation	Neural network
Evaluation	Squared error
Optimization	Gradient descent
Algorithms	Backpropagation

Evolutionaries

- Natural selection is the mother of all learning
- Evolutionary algorithms
- Crossover, mutation
- Can learn structure, wide hypothesis space
- Needs a way to 'fill' the structure

Representation	Genetic programs (often trees)
Evaluation	Fitness function
Optimization	Genetic search
Algorithms	Genetic programming (crossover, mutation)

Bayesians

- Learning is a form of uncertain inference
- Graphical models, Gaussian processes, HMMs, Kalman filter
- Uses Bayes theorem to incorporate new evidence into our beliefs
- Can deal with noisy, incomplete, contradictory data
- Depends on the prior
- Hard to do unite logic and probability

Representation	Graphical models, Markov networks
Evaluation	Posterior probability
Optimization	Probabilistic inference
Algorithms	Bayes theorem and derivates

Analogizers

- You are what you resemble
- Recognizes similarities between situations and infers other similarities
- k-Nearest Neighbor, Support Vector Machines
- Transfer solution from previous situations to new situations
- Hard to do rules and structure

Representation	Memory, support vectors
Evaluation	Margin
Optimization	Kernel machines
Algorithms	k-Nearest Neighbor, Support Vector Machines

The Master Algorithm?

Tribe	**Problem**	**Solution**
Symbolists	Knowledge composition	Inverse deduction
Connectionists	Credit assignment	Backpropagation
Evolutionaries	Structure discovery	Genetic programming
Bayesians	Uncertainty	Probabilistic inference
Analogizers	Similarity	Kernel machines

- Representation: The hypothesis space.
  - Probabilistic logic
  - Weighted formulas $\rightarrow$ Distribution over states
- Evaluation: How to choose one hypothesis over the other?
  - Posterior probability
  - User-defined objective function
- Optimization: How do we search the hypothesis space?
  - Formula discovery: Genetic programming
  - Weight learning: Backpropagation

Elegant/Extensible/Expressive/Efficient/Educable/Evolvable?

Learn to Learn

1. Good Old-Fashioned AI
   - Handcraft predictions
   - Learn nothing
2. Shallow Learning
   - Handcraft features
   - Learn predictions
3. Deep Learning
   - Handcraft algorithm (optimiser, target, architecture, ...)
   - Learn features and predictions end-to-end
4. Meta Learning
   - Handcraft nothing
   - Learn algorithm and features and predictions end-to-end

Can Machines Think?

- Theological objections.
- Argument from informality of behavior. — Human behavior is far too complex to be captured by any simple set of rules./Learning from experience.
- Argument from incompleteness theorems. — No formal system incl. AIs, but only humans can "see" that Gödel's unprovable sentence is true./Lucas cannot consistently assert that this sentence is true.
- Machines can't be conscious or feel emotions. — Reductionism doesn't really answer the question: why can't machines be conscious or feel emotions?
- Machines don't have Human Quality X.
- Machines just do what we tell them to do. — Maybe people just do what their neurons tell them to do.
- Machines are digital. Mental states can emerge from neural substrate only. — Only the functionality/behavior matters.
- Non-Computable Physics & Brains.

Brain Dissection

- The "brain in a vat" experiment: (no) real experience.
- The brain prosthesis experiment:
  - Replacing some neurons in the brain by functionally identical electronic prostheses would neither effect external behavior nor internal experience of the subject.
  - Successively replace one neuron after the other until the whole brain is electronic.

Fermi Paradox — Where are the aliens?

- There are none, i.e. we're all alone.
- We can't detect them because...
  - we're too primitive or too far apart
  - there are predators or all fear them
  - we're lied to, live in a simulation
  - ...

Why do we need to align machine agents?

- **Goal orthogonality.**
  Intelligence and final goals are orthogonal: Any utility function can be combined with a powerful epistemology and decision theory.
- **Instrumental convergence.**
  Different long-term goals imply similar short-term strategies.
  - Self-preservation
  - Retention of goals through time
  - Cognitive enhancement
  - Technological perfection
  - Resource acquisition
  - ...
- **Capability gain.**
  There are potential ways for artificial agents to greatly gain in cognitive power and strategic options.
- **Alignment difficulty.**
  It's hard to transmit our values to AI systems or avert adversarial incentives.

Asimov's Laws of Robotics

- **Law Zero**
  A robot may not injure humanity, or, through inaction, allow humanity to come to harm.
- **Law One**
  A robot may not injure a human being or, through inaction, allow a human being to come to harm, unless this would violate a higher order law.
- **Law Two**
  A robot must obey orders given it by human beings except where such orders would conflict with a higher order law.
- **Law Three**
  A robot must protect its own existence as long as such protection does not conflict with a higher order law.

An Extended Set of the Laws of Robotics

- **The Meta-Law**
  A robot may not act unless its actions are subject to the Laws of Robotics.
- **Law Zero**
  A robot may not injure humanity, or, through inaction, allow humanity to come to harm.
- **Law One**
  A robot may not injure a human being, or, through inaction, allow a human being to come to harm, unless this would violate a higher-order Law.
- **Law Two**
  1. A robot must obey orders given it by human beings, except where such orders would conflict with a higher-order Law.
  2. A robot must obey orders given it by superordinate robots, except where such orders would conflict with a higher-order Law.
- **Law Three**
  1. A robot must protect the existence of a superordinate robot as long as such protection does not conflict with a higher-order Law.
  2. A robot must protect its own existence as long as such protection does not conflict with a higher-order Law.
- **Law Four**
  A robot must perform the duties for which it has been programmed, except where that would conflict with a higher-order law.
- **The Procreation Law**
  A robot may not take any part in the design or manufacture of a robot unless the new robot's actions are subject to the Laws of Robotics.

Ethical Concerns

- Is it morally justified to create intelligent systems with these constraints?
- Would it be possible to do so?
- Should intelligent systems have free will? Can we prevent them from having free will? What could it mean for a machine to have its own goals? Do we have a kind of freedom machines could never have?
- Will intelligent systems have consciousness?
- If they do, will it drive them insane to be constrained by artificial ethics placed on them by humans?
- If intelligent systems develop their own ethics and morality, will we like what they come up with?

Machine Ethics

- People might lose their jobs to automation.
- People might have too much (or too little) leisure time.
- People might lose their sense of being unique.
- People might lose some of their privacy rights.
- The use of AI systems might result in a loss of accountability.
  — Who is responsible if a physician follows the advice of a medical expert system, whose diagnosis turns out to be wrong?
- The success of AI might mean the end of the human race.

What If We Do Succeed?

- Natural selection is replaced by artificial evolution.
  — AI systems will be our mind children.
- Once a machine surpasses the intelligence of a human it can design even smarter machines.
- This will lead to an intelligence explosion and a technological singularity at which the human era ends.
- Prediction beyond this event horizon will be impossible.
- Alternative 1: We keep the machines under control.
  - Capability control (limiting what the system can or does do).
  - Motivation selection (controlling what the system wants to do).
- Alternative 2: Humans merge with or extend their brain by AI.

Single-Shot Situation

Our first superhuman AI must be a safe one for we may not get a second chance!

Singularity

- Speed Explosion (Time).
  — Computing speed doubles every two subjective years of work.
- Population Explosion (Quantitative).
  — Computing costs halve for a certain amount of work.
- Intelligence Explosion (Qualitative).
  — Proportionality Thesis: An increase in intelligence leads to similar increases in the capacity to design intelligent systems.

Capability Control Methods

- Boxing: the system can only act through restricted channels.
  — The AI could persuade someone to free it from its box.
- Incentives: access to other AIs, cryptographic reward tokens.
- Stunting: imposing constraints on the system's cognitive abilities
- Tripwires: diagnostic tests run periodically to check for dangerous activity, with shutdown a consequence of detection.

Motivation Selection Methods

- Direct specification of motivations
  - Rule-based methods: give the machine a set of rules that define its final goals.
  - Direct consequentialist methods: specify some measure that is to be maximised. (e.g. human happiness.)
- Augmentation
  - Start with an AI with human-level intelligence, that has an acceptable motivation system: then enhance its cognitive faculties to make it superintelligent.
- Indirect normativity
  - Rather than specifying a normative standard directly, we specify a process for deriving a standard. We then build the system so it is motivated to carry out this process.
  - Value learning. — inverse reinforcement learning. — wireheading.

Prisoner's Dilemma

- Difficult to prevent arms races.
- The winner takes all (of what remains).
- Arms races are dangerous because parties sacrifice safety for speed!
  — When headed the wrong way, the last thing we need is progress.

International Cooperation

In face of uncertainty, cooperation is robust!

Why should I care about the world when I am dead and gone? I want it to go fast, damn it! This increases the chance I have of experiencing a more technologically advanced future.

AI Alignment

- Important to ensure it's aligned with our interests
  - But how do we specify beneficial goals?
  - How do we make sure system actually pursues them?
  - How do we correct the system if we get it wrong?
- Want solid theoretical understanding of problem & solution
  - What is correct reasoning and decision making?
  - Probability theory, decision theory, game theory, statistical learning theory, Bayesian networks, formal verification, ...

Technical Research Questions

1. Reliable self-modification
2. Logical uncertainty (reasoning without logical omniscience)
3. Reflective stability of decision theory
4. Decision theory for Newcomb-like problems
5. Corrigibility (accepting modifications)
6. The shutdown problem
7. Value loading
8. Indirect specification of decision theory
9. Domesticity (goal specification for limited impact)
10. The competence gap
11. Weighting options or outcomes for variance-normalizing solution to moral uncertainty
12. Program analysis for self-improvement
13. Reading values and beliefs of AIs
14. Pascal's mugging
15. Infinite ethics
16. Mathematical modelling of intelligence explosion

Before the prospect of an intelligence explosion, we humans are like children playing with a bomb. Such is the mismatch between the power of our play-thing and the immaturity of our conduct. Superintelligence is a challenge for which we are not ready now and will not be ready for a long time. We have little idea when the detonation will occur, though if we hold the device to our ear we can hear a faint ticking sound.

— Nick Bostrom

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Contents

1. History of AI
2. Philosophy of Induction
   - History
   - How to Choose the Prior?
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Hume & Russell

Proposition (Hume)

Induction is just a mental habit, and necessity is something in the mind and not in the events.

Leibniz-Wittgenstein-Goodman

Proposition (Leibniz)

Since for any finite number of points there are always infinitely many curves going through them, any finite set of data is compatible with infinitely many inductive generalizations.

Proposition (Wittgenstein)

Since any finite course of action is in accord with infinitely many rules, no universal rule can be learned by examples.

Proposition (Goodman)

All emeralds discovered till 2050 are green, and blue thereafter.

Mill — Homogeneous Universe

Proposition (Mill)

Induction can be turned into a deduction, by adding principles about the world (such as 'the future resembles the past', or 'space-time is homogeneous').

Homogeneous?

Problem

$1, 3, 5, 7, 9, 11, 13, 15, ?$

Solution

$$2n - 1 \qquad (17?)$$

$$2n - 1 + \prod_{i=1}^{8}(n - i) \qquad (17+8!?)$$

$$\vdots \qquad (??)$$

Epicurus vs Occam

Proposition (Epicurus)

If more than one hypothesis is consistent with the observations, keep them all.

Proposition (Occam's Razor)

Among the hypotheses that are consistent with the observed phenomena, select the simplest one.

- Entities should not be multiplied beyond necessity.
- Wherever possible, logical constructions are to be substituted for inferred entities.
- It is vain to do with more what can be done with fewer.

Why Simplicity? — Gestalt Psychology

Figure: Gestalt Psychology

Why Simplicity?

God does not play dice.
God always takes the simplest way.
Subtle is the Lord, but ***malicious*** *He is not.*
The most incomprehensible thing about the world is that it is comprehensible.
What really interests me is whether God could have created the world any differently; in other words, whether the requirement of logical simplicity admits a margin of freedom.
When I am judging a theory, I ask myself whether, if I were God, I would have arranged the world in such a way.

— *Einstein*

- Principle of least/stationary action
- Noether's theorem
- ...

Why Simplicity?

program	$\xrightarrow{\textit{Computer}}$	*output*
axioms	$\xrightarrow{\textit{Deduction}}$	*theorems*
scientific theory	$\xrightarrow{\textit{Calculations}}$	*experimental data*
encoded message	$\xrightarrow{\textit{Decoder}}$	*original message*
software	$\xrightarrow{\textit{Universal Constructor}}$	*physical system*
DNA	$\xrightarrow{\textit{Pregnancy}}$	*organism*
Ideas	$\xrightarrow{\textit{Mind of God}}$	*Universe*

Popper — The Logic of Scientific Discovery

Proposition (Popper)

- *A single observational event may prove hypotheses wrong, but no finite sequence of events can verify them correct.*
- *Induction is theoretically unjustifiable and becomes in practice the choice of the simplest generalization that resists falsification.*
- *The simpler a hypothesis, the easier it is to be falsified.*
- *Falsifiability is as subjective as simplicity, there is no objective criterion.*

Kaynes $\Longrightarrow$ Carnap

- Assign to inductive generalizations probabilities that should converge to 1 as the generalizations are supported by more and more independent events.

  — Kaynes

- Observational events provide, if not proofs, at least positive confirmations of scientific hypotheses. Chose the generalization that confirm more evidence.

  — Carnap

Philosophy of Induction

What is learnable? How to learn?
How can we know that what we learned is true?

History

Possible Worlds/Hypothesis (Epicurus/Leibniz)

$+$

Homogeneous Universe(s) (Mill/Turing)

$+$

Simplicity Criterion (Occam/Kolmogorov)

$+$

Prior Belief (Carnap/Solomonoff)

$+$

Update Belief (Bayes)

$\Downarrow$

Convergence to Truth

$$P(h|e) = \frac{P(e|h)P(h)}{\sum_{h\in\mathcal{H}} P(e|h)P(h)} \xrightarrow{\ell(e)\to\infty} 1$$

MDL vs Bayesian Mixture

When solving a problem of interest, do not solve a more general problem as an intermediate step.

Bayesianism

Theorem (Convergence Theorem)

$$\sum_{t=1}^{n} \mathbb{E}_\mu \left[\left(\sqrt{\frac{\rho(a|x_{<t})}{\mu(a|x_{<t})}} - 1 \right)^2 \right] \leq \sum_{t=1}^{n} \mathbb{E}_\mu \left[\sum_{a \in \mathcal{X}} \left(\sqrt{\rho(a|x_{<t})} - \sqrt{\mu(a|x_{<t})} \right)^2 \right] \leq D_n(\mu \| \rho)$$

$$\sum_{t=1}^{n} \mathbb{E}_\mu \left[\sum_{a \in \mathcal{X}} \left(\rho(a|x_{<t}) - \mu(a|x_{<t}) \right)^2 \right] \leq D_n(\mu \| \rho)$$

$$\frac{1}{2n} \left(\sum_{t=1}^{n} \mathbb{E}_\mu \left[\sum_{a \in \mathcal{X}} \left| \rho(a|x_{<t}) - \mu(a|x_{<t}) \right| \right] \right)^2 \leq D_n(\mu \| \rho)$$

where

$$D_n(\mu \| \rho) := \mathbb{E}_\mu \left[\ln \frac{\mu(x_{1:n})}{\rho(x_{1:n})} \right]$$

Theorem

$$\xi = \operatorname*{argmin}_{\rho} \mathbb{E}_w \left[D(\mu \| \rho) \right] \quad \textit{where} \quad \xi(x) := \sum_{\nu \in \mathcal{M}} w_\nu \nu(x)$$

Bayesian Decisions

Suppose $Loss(x_t, y_t) \in [0,1]$

$$y_t^{\Lambda_\rho}(x_{<t}) \;:=\; \arg\min_{y_t} \sum_{x_t} \rho(x_t|x_{<t}) Loss(x_t, y_t)$$

$$L^{\Lambda_\rho}(x_{<t}) \;:=\; \mathbb{E}_\mu \left[Loss\left(x_t, y_t^{\Lambda_\rho}\right) \middle| x_{<t} \right]$$

$$L_n^{\Lambda_\rho} \;:=\; \sum_{t=1}^{n} \mathbb{E}_\mu \left[L^{\Lambda_\rho}(x_{<t}) \right]$$

Theorem

$$\left(\sqrt{L_n^{\Lambda_\xi}} - \sqrt{L_n^{\Lambda_\mu}}\right)^2 \leq 2 \sum_{t=1}^{n} \mathbb{E}_\mu \left[\sum_{a \in \mathcal{X}} \left(\sqrt{\xi(a|x_{<t})} - \sqrt{\mu(a|x_{<t})} \right)^2 \right] \leq 2 D_n(\mu \| \xi)$$

$$L_n^{\Lambda_\xi} - L_n^{\Lambda_\mu} \leq 2 D_n(\mu \| \xi) + 2\sqrt{L_n^{\Lambda_\mu} D_n(\mu \| \xi)}$$

Problem

How to choose the model class and prior?

- choose the smallest model class that will contain the true environment.
- choose the priors that best reflect a rational a-priori belief in each of these environments.
  1. Convergence of Bayesian mixture to true environment.
  2. Confirmation of "the sun will always rise".
  3. Invariance Criterion.
     reparametrization & regrouping invariant.

Contents

1. History of AI
2. Philosophy of Induction
   - History
   - How to Choose the Prior?
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Invariance Criterion

By applying some principle to a parameter θ we get prior $w(\theta)$. If we consider some new parametrization θ' via $f\colon \theta \mapsto \theta'$, then we get a prior $\tilde{w}(\theta')$ by transforming the original prior via f.
for discrete class $\mathcal{M}$,

$$\tilde{w}(\theta') := \sum_{\theta : f(\theta) = \theta'} w(\theta)$$

for continuous parametric class $\mathcal{M}$,

$$\tilde{w}(\theta') := \int \delta(f(\theta) - \theta') w(\theta) \mathrm{d}\theta \qquad \text{(Dirac-delta)}$$

Regrouping-invariant:

$$\tilde{w}(\theta') = w'(\theta')$$

where $w'(\theta')$ is obtained by applying the same principle to the new parametrization.
We say the principle is reparametrization-invariant when f is bijective.

How to Assign Prior? Indifference Principle/MaxEnt

How to Assign Prior?

- The principle of indifference. Assume $w(\theta) = 1$ and $\theta' = \sqrt{\theta}$.

$$\tilde{w}(\theta') = \int_0^1 \delta\left(\sqrt{\theta} - \theta'\right) w(\theta) \mathrm{d}\theta = 2\sqrt{\theta} \neq w'(\theta')$$

- Maximize the entropy subject to some constraints provided by empirical data or considerations of symmetry, probabilistic laws, and so on.
- Occam's razor.

How to confirm “All Ravens are Black”?

Problem 1 — All Ravens are Black $\theta = 1$

Suppose θ is the percentage of ravens that are black.
"All ravens are black" $\equiv$ $\theta = 1$. or, 1^∞ or, $\forall x\colon R(x) \to B(x)$

$$P(\theta = 1) = \int_1^1 w(\theta)\mathrm{d}\theta = 0 \qquad (0 \text{ prior})$$

$$\Downarrow$$

$$P(\theta = 1|1^n) = \frac{P(1^n|\theta = 1)P(\theta = 1)}{P(1^n)} = 0 \qquad (\times)$$

Problem 2 — The Sun will always Rise 1^∞

Indifference Principle

$$\left.\begin{aligned} \int_0^1 w(\theta)\mathrm{d}\theta = 1 \\ \forall \theta, \theta' : w(\theta) = w(\theta') \end{aligned}\right\} \implies \forall \theta : w(\theta) = 1$$

The sun will rise tomorrow. ✓

$$P(x) = \int_0^1 P(x|\theta) w(\theta) \mathrm{d}\theta \implies P(1|1^n) = \frac{n+1}{n+2}$$

The sun will always rise. ×

$$P(1^\infty|1^n) = \lim_{k\to\infty} P(1^k|1^n) = \lim_{k\to\infty} \frac{n+1}{n+k+1} = 0$$

Solution 1 — Soft Hypothesis — No absolute truth!

$$H_\varepsilon = \{\theta : \theta \in (1-\varepsilon, 1]\}$$

$$P(H_\varepsilon) = \int_{1-\varepsilon}^{1} w(\theta)\mathrm{d}\theta = \varepsilon > 0$$

$$\begin{aligned}
P(H_\varepsilon|1^n) &= \int_{1-\varepsilon}^{1} w(\theta|1^n)\mathrm{d}\theta \\
&= \int_{1-\varepsilon}^{1} \frac{P(1^n|\theta)w(\theta)}{P(1^n)}\mathrm{d}\theta \\
&= \int_{1-\varepsilon}^{1} (n+1)\theta^n \mathrm{d}\theta \\
&= \theta^{n+1}\Big|_{1-\varepsilon}^{1} \\
&= 1-(1-\varepsilon)^{n+1} \xrightarrow{n\to\infty} 1
\end{aligned}$$

Solution 2 — ad hoc

$$w(\theta) := \frac{1}{2}(1+\delta(1-\theta))$$

Dirac-delta sifting property: $\displaystyle\int f(\theta)\delta(\theta-a)\,\mathrm{d}\theta = f(a)$

$$\begin{aligned}
P(x) &= \int_0^1 P(x|\theta)w(\theta)\mathrm{d}\theta \\
&= \int_0^1 \theta^s(1-\theta)^f \cdot \frac{1}{2}(1+\delta(1-\theta))\mathrm{d}\theta \\
&= \frac{1}{2}\int_0^1 \theta^s(1-\theta)^f\,(1+\delta(\theta-1))\mathrm{d}\theta \\
&= \frac{1}{2}\left(\frac{s!f!}{(s+f+1)!} + 1^s\cdot(1-1)^f\right) && \text{[sifting property]} \\
&= \frac{1}{2}\left(\frac{s!f!}{(s+f+1)!} + \delta_{f,0}\right)
\end{aligned}$$

Solution 2 — ad hoc

$$P(1^\infty|1^n) = \lim_{k\to\infty} \frac{P(1^{n+k})}{P(1^n)} = \lim_{k\to\infty} \frac{\frac{1}{2}\left(\frac{(n+k)!0!}{(n+k+1)!}+1\right)}{\frac{1}{2}\left(\frac{n!0!}{(n+1)!}+1\right)} = \frac{n+1}{n+2} \xrightarrow{n\to\infty} 1$$

$$P(\theta \geq a) = \int_a^1 \frac{1}{2}(1+\delta(\theta-1))\mathrm{d}\theta = 1 - \frac{1}{2}a$$

$$P(\theta = 1) = \frac{1}{2}$$

$$P(\theta = 1|1^n) = \frac{P(1^n|\theta=1)P(\theta=1)}{P(1^n)} = \frac{1\cdot\frac{1}{2}}{\frac{1}{2}\left(\frac{n!0!}{(n+1)!}+1\right)} = \frac{n+1}{n+2} \xrightarrow{n\to\infty} 1$$

Why $\theta = 1$ special?

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Natural Wish List

- (computability) $P_n(\varphi)$ is computable.
- (convergence) $P(\varphi) = \lim_{n\to\infty} P_n(\varphi)$
- (coherent limit) $P(\varphi \wedge \psi) + P(\varphi \vee \psi) = P(\varphi) + P(\psi)$
- (non-dogmatism) If $\nvdash \varphi$ then $P(\varphi) < 1$, and if $\nvdash \neg\varphi$ then $P(\varphi) > 0$.

Unary Pure Inductive Logic

- $\mathscr{L}$ contains countable constants $\mathscr{C}$ and m unary predicates.
- $\mathscr{R} = \{R_1, R_2, \ldots, R_m\}$ with no function symbols nor equality.
- $Q_i := \bigwedge_{j=1}^{m} \pm R_j$ for $1 \leq i \leq 2^m =: r$.
- $\mathscr{Q} = \{Q_1, \cdots, Q_r\}$ is a r-fold classification system of some Universe with domain $\mathscr{C}$.

Definition (Probability on Sentences)

A probability on sentences is a non-negative function $w \colon \mathscr{S} \to [0,1]$ s.t.

P_1. $\vDash \psi \implies w(\psi) = 1$

P_2. $\psi_1 \vDash \neg\psi_2 \implies w(\psi_1 \vee \psi_2) = w(\psi_1) + w(\psi_2)$

P_3. $w\big(\exists x \psi(x)\big) = \lim_{n\to\infty} w\left(\bigvee_{i=1}^{n} \psi(a_i)\right)$

Properties

Theorem

i. $w(\neg\varphi) = 1 - w(\varphi)$
ii. $\vDash \neg\varphi \implies w(\varphi) = 0$
iii. *The following are equivalent:*
   a. $w(\varphi) = 1 \implies \vDash \varphi$
   b. $w(\varphi) = 0 \implies \vDash \neg\varphi$
iv. $\varphi \vDash \psi \implies w(\varphi) \leq w(\psi)$
v. $\vDash \varphi \leftrightarrow \psi \implies w(\varphi) = w(\psi)$
vi. $w(\varphi) + w(\psi) = w(\varphi \wedge \psi) + w(\varphi \vee \psi)$

Theorem (Extension Theorem)

For any probability function over quantifier-free sentences $w\colon \mathscr{S} \to [0,1]$ *satisfying* P_1, P_2, w *has an unique extension to* $\overline{w}\colon \mathscr{S} \to [0,1]$ *satisfying* P_1, P_2, P_3.

Possible Worlds

- **state description** $\bigwedge_{i=1}^{n} Q_{h_i}(a_i)$

  where $h\colon \mathscr{C} \to \mathscr{Q}$
- **structure description** $\{n_i\}_{i=1}^{r}$

  where $n_i := \sum_{j=1}^{n} [\![h_j = i]\!]$
- **rank description** $\{m_i\}_{i=0}^{n}$

  where $m_i := \sum_{j=1}^{r} [\![n_j = i]\!]$

  Obviously, $\sum_{i=1}^{n} i \cdot m_i = n$ and $\sum_{i=0}^{n} m_i = r$.

Indifference Principle

- A All state descriptions have equal weight.
- B All structure descriptions have equal weight.
- C Each nonempty subset of the alphabet is equally likely.
- D Each nonzero cardinality is equally likely.
- E All rank descriptions have equal weight.

Given n individuals, there are r^n possible state descriptions,

$$\left|\left\{(n_1, \ldots, n_r) \colon \sum_{i=1}^{r} n_i = n\right\}\right| = \binom{n+r-1}{r-1}$$

possible structure descriptions, and

$$p(n,r) := \left|\left\{(m_0, \ldots, m_n) \colon \sum_{i=1}^{n} i \cdot m_i = n \ \ \& \ \ \sum_{i=0}^{n} m_i = r \ \ \& \ \ \forall i \colon m_i \geq 0\right\}\right|$$

possible rank descriptions.

(A) State Description ×

According to (A),

$$m^\dagger\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = \frac{1}{r^n}$$

$$c^\dagger\left(Q_j(a_{n+1}) \,\middle|\, \bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = \frac{m^\dagger\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i) \wedge Q_j(a_{n+1})\right)}{m^\dagger\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right)} = \frac{1}{r}$$

(B) Structure Description ✓

According to (B),

$$m^*(n_1, \ldots, n_r) = \frac{1}{\binom{n+r-1}{r-1}}$$

Structure Description $(n_1, \ldots, n_r)$ corresponds to $\binom{n}{n_1, \ldots, n_r}$ State Descriptions.

$$m^*\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = \frac{m^*(n_1, \ldots, n_r)}{\binom{n}{n_1, \ldots, n_r}} = \frac{1}{\binom{n+r-1}{r-1}\binom{n}{n_1, \ldots, n_r}}$$

Sometimes we write $m^*(h_{1:n}) := m^*\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right)$ for short.

Carnap's Degree of Confirmation

$$c^*\left(Q_j(a_{n+1}) \,\middle|\, \bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = \frac{m^*\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i) \wedge Q_j(a_{n+1})\right)}{m^*\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right)} = \frac{n_j + 1}{n + r}$$

frequency — independent identical distribution(i.i.d)
extension

$$c^*(\varphi)$$

(C,D,E)

According to (C),

$$m^{\$}(h_{1:n}) = \frac{1}{\left(\sum_{i=1}^{\min\{r,n\}} \binom{r}{i}\right) \binom{n-1}{r-m_0-1} \binom{n}{n_1,\ldots,n_r}}$$

According to (D),

$$m^{\#}(h_{1:n}) = \frac{1}{\min\{r,n\} \binom{r}{r-m_0} \binom{n-1}{r-m_0-1} \binom{n}{n_1,\ldots,n_r}}$$

According to (E),

$$m^{\tau}(h_{1:n}) = \frac{1}{\binom{n}{n_1,\ldots,n_r} \binom{r}{m_0,\ldots,m_n} p(n,r)}$$

Rank Description $(m_0, \ldots, m_n)$ corresponds to $\binom{r}{m_0,\ldots,m_n}$ Structure Descriptions.

What is the right w?

- Constant Exchangeability Principle.
  For any permutation σ of $\mathbb{N}^+$,

$$w(\psi(a_1, \ldots, a_n)) = w(\psi(a_{\sigma(1)}, \ldots, a_{\sigma(n)})) \tag{Ex}$$

- Atom Exchangeability Principle.
  For any permutation τ of $\{1, 2, \ldots, r\}$,

$$w\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = w\left(\bigwedge_{i=1}^{n} Q_{\tau(h_i)}(a_i)\right) \tag{Ax}$$

- Sufficientness Postulate.

$$w\left(Q_j(a_{n+1}) \,\middle|\, \bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right) = f_j(n_j, n) \tag{SP}$$

What is the right w?

Principle Ex asserts that $w\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right)$ depends only on the vector $\left\langle n_{h_i} : 1 \leq i \leq n\right\rangle$, so that it is independent on the order of observing the individuals, while in the presence of Ex, principle Ax asserts that $w\left(\bigwedge_{i=1}^{n} Q_{h_i}(a_i)\right)$ depends only on $\{n_i : 1 \leq i \leq r\}$, and $w(Q_i(a_1)) = 1/r$ for all $1 \leq i \leq r$.

Carnap's λ-continuum

Theorem

Suppose language $\mathscr{L}$ has at least two predicates i.e. $m \geq 2$, then the probability function w on $\mathscr{L}$ satisfies Ex, SP iff $w = c_\lambda$ for some $0 \leq \lambda \leq \infty$.

Namely,

$$f_i(n_i, n) = \frac{n_i + \lambda\gamma_i}{n + \lambda}$$

where $\gamma_i = f_i(0,0)$ and $\lambda = \dfrac{f_i(0,1)}{f_i(0,0) - f_i(0,1)}$.

By adding Ax, $\forall i : \gamma_i = \dfrac{1}{r}$.

Shortcoming 1 — All Ravens are Black? ×

$$c^*\left(\forall x\,(R(x) \to B(x))\right) \leq \lim_{n\to\infty} \prod_{i=0}^{n-1} \frac{i+r-1}{i+r} = 0$$

Convergence Speed? Yes and No!

$$\prod_{n\geq 1} a_n = 0 \iff \sum_{n\geq 1}(1-a_n) = \infty \quad \text{for} \quad \forall n\colon 0 < a_n \leq 1$$

Shortcoming 2 — No-Free-Lunch for Carnap!

Strengthened "Hume" — Wolpert & Macready 1997, Igel & Toussaint 2004:

No-Free-Lunch Theorem!

All state descriptions with the same structure description have equal weight!

Block Uniform

No Free Lunch Theorem — Strengthened "Hume"

Theorem (No Free Lunch Theorem)

If and only if the probability distribution P is block uniform, i.e.

$$\forall f, g \in \mathcal{Y}^{\mathcal{X}} : \forall y \in \mathcal{Y} \left(\left| f^{-1}(y) \right| = \left| g^{-1}(y) \right| \right) \implies P(f) = P(g)$$

then for any two algorithms A, A', any value $k \in \mathbb{R}$, any $m \in \{1, \ldots, |\mathcal{X}|\}$, and any performance measure L,

$$\sum_{f \in \mathcal{Y}^{\mathcal{X}}} P(f) \left[\!\left[k = L\left(T_m^y(A, f)\right) \right]\!\right] = \sum_{f \in \mathcal{Y}^{\mathcal{X}}} P(f) \left[\!\left[k = L\left(T_m^y(A', f)\right) \right]\!\right]$$

where $T_n := \langle (x_1, f(x_1)), \ldots, (x_n, f(x_n)) \rangle$, $T_n^x := \langle x_1, \ldots, x_n \rangle$, $T_n^y := \langle f(x_1), \ldots, f(x_n) \rangle$ and $A : T_n \mapsto x_{n+1} \in \mathcal{X} \setminus T_n^x$.

equally well and equally poorly

No learning is possible without some prior knowledge!

No-Free-Lunch for Pure Inductive Logic

- No learning is possible for $c^\dagger$. It seems possible to learn with c^* and c_λ. Unfortunately, No-Free-Lunch Theorem!
- Take $\mathcal{X} := \mathscr{C}, \mathcal{Y} := \mathscr{Q}$ in the No-Free-Lunch Theorem. The state description $h\colon \mathscr{C} \to \mathscr{Q}$ can be taken as a *classification* function.
- Then "all state descriptions $h\colon \mathscr{C} \to \mathscr{Q}$ with the same structure description $\{n_i : 1 \le i \le r\}$ have equal weight" is block uniform!
- Define induction algorithm $A(h_{1:n}) := \operatorname*{argmax}_j c^* \left(Q_j(a_{n+1}) | h_{1:n}\right)$, and loss function $L\left(A, n, h\right) := [\![A(h_{1:n}) \neq h_{n+1}]\!]$. Then for any A',

$$\sum_{h \in \mathcal{Y}^{\mathcal{X}}} m^*(h_{1:n}) L\left(A, n, h\right) = \sum_{h \in \mathcal{Y}^{\mathcal{X}}} m^*(h_{1:n}) L\left(A', n, h\right)$$

- Similarly for rank description (E), which is related to Good-Turing estimate. And similarly for Ristad's methods $(C)(D)$.

Time Series and Solomonoff Induction

- However, if we take time into consideration, define

$$M(h_{1:n}) := \sum_{p:U(p)=h_{1:n}*} 2^{-\ell(p)}$$

then we can get free lunch with M', since M' biases non-random state descriptions and is not block uniform. where

$$M'(\epsilon) := 1$$

$$M'(h_{1:n}) := M'(h_{<n}) \frac{M(h_{1:n})}{\sum_{Q \in \mathcal{Q}} M(h_{<n}Q)}$$

and we can extend M' from state descriptions to sentences.

- Besides, by c^* the probability of "the sun will rise tomorrow" is $\frac{n+1}{n+2}$, but c^* fails to confirm "the sun will always rise", while M' can confirm it.

$$\lim_{n\to\infty} M'(\text{the sun will always rise}|\text{the sun rises in the first } n \text{ days}) = 1$$

PAC (Probably Approximately Correct) Learning

Definition (PAC-Learnability)

A hypothesis space $\mathcal{H} \subset 2^{\mathcal{X}}$ is PAC-learnable if there exists a sample complexity function $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ and a learning algorithm A with the following property:

- for every $\varepsilon, \delta \in (0,1)$
- for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labeling function $f \colon \mathcal{X} \to \{0,1\}$

when running A on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. training samples S generated by $\mathcal{D}$ and labeled by f, the algorithm A returns a hypothesis $A(S) \in \mathcal{H}$ s.t.

$$P_{S \sim \mathcal{D}^m}\Big(P_{x \sim \mathcal{D}}\big(A(S)(x) \neq f(x)\big) > \varepsilon\Big) < \delta$$

$$\text{No-Free-Lunch} \implies m_{2^{\mathcal{X}}}(\varepsilon, \delta) = \infty$$

Agnostic PAC Learning

Definition (Agnostic PAC-Learnability)

A hypothesis space $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ is agnostic PAC-learnable under a class Δ of distributions with respect to $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ and a loss function $\ell \colon \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$, if there exists a sample complexity function $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ and a learning algorithm A with the following property:

- for every $\varepsilon, \delta \in (0,1)$
- for every distribution $\mathcal{D} \in \Delta$ over $\mathcal{Z}$

when running A on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. training samples S generated by $\mathcal{D}$, the algorithm A returns a hypothesis $A(S) \in \mathcal{H}$ s.t.

$$P_{S \sim \mathcal{D}^m}\left(L_{\mathcal{D}}(A(S)) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) > \varepsilon\right) < \delta$$

where $L_{\mathcal{D}}(h) := \mathbb{E}_{z \sim \mathcal{D}}\left[\ell(h, z)\right]$.

PAC-learnable under $\Delta := \{\mathcal{D} \colon \mathcal{D} \overset{\times}{\lesssim} \xi\} \iff$ PAC-learnable under $\xi(x) := \sum_{\nu \in \mathcal{M}} 2^{-K(\nu)} \nu(x)$.

VC-Dimension

Definition (Shattering)

A hypothesis space $\mathcal{H} \subset 2^{\mathcal{X}}$ shatters a set $C \subset \mathcal{X}$ if $\mathcal{H}\restriction_C = 2^C$.

Definition (VC-Dimension)

$\mathrm{VC}(\mathcal{H}) := \sup\left\{|C| : \mathcal{H} \text{ shatters } C\right\}$.

If someone can explain every phenomena, her explanations are worthless.

Theorem

If $\mathrm{VC}(\mathcal{H}) = \infty$, then $\mathcal{H}$ is not PAC-learnable.

Theorem (Fundamental Theorem of PAC Learning)

$\mathcal{H}$ is PAC-learnable iff $\mathrm{VC}(\mathcal{H}) < \infty$. Indeed, there exists C_1, C_2 s.t.

$$C_1 \frac{\mathrm{VC}(\mathcal{H}) + \log(1/\delta)}{\varepsilon} \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq C_2 \frac{\mathrm{VC}(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}$$

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Formal Learning Theory

- Carnap's inductive logic is a design for a 'learning machine' that can extrapolate certain kinds of empirical regularities from the data with which it is supplied, and the task of inductive logic is to construct a 'universal learning machine'.
- If there is such a thing as a correct definition of 'degree of confirmation' which can be fixed once and for all, then a machine that predicts in accordance with it would be a cleverest possible learning machine.
- Either there are better and better 'degree of confirmation' functions, but no 'best possible', or there is a 'best possible' but it is not computable by a machine.

Formal Learning Theory

- Putnam 1963
- Gold 1967 $\hat{f}(n+1) = g(\langle f(0), \ldots, f(n)\rangle)$ $\varphi_{\lim_{n\to\infty} g(\langle f(0),\ldots,f(n))} = f$
- Solomonoff 1960

Ray Solomonoff

Figure: Solomonoff

- 0101010101010101010101010101010101
- 0101001101111011011101010100000 1
- 0010010000111111011010101000 1000

$\{\pi\}$

1. What is regularity/pattern/law/principle/model/hypothesis/theory?
2. What is phenomenon/data/experience?
3. What is randomness/noise?
4. What is typicalness/unpredictability/incompressibility?
5. What is simplicity/complexity?
6. What is learning?
7. What is beauty/interesting/curiosity/novelty/surprise/creativity?
8. What is intelligence?

Text $\Longrightarrow$ Meaning?

numeral $\xrightarrow{\text{short algorithm}}$ value

汝有田舍翁，家资殷盛，而累世不识“之”“乎”。一岁，聘楚士训其子。楚士始训之搦管临朱，书一画，训曰：“一”字。书二画，训曰：“二”字。书三画，训曰：“三”字。其子辄欣欣然掷笔，归告其父曰：“儿得矣！儿得矣！可无烦先生，重费馆谷也，请谢去。”其父喜从之，具币谢遣楚士。

逾时，其父拟征召姻友万氏者饮，令子晨起治状，久之不成。父趣之。其子恚曰：“天下姓字多矣，奈何姓万？自晨起至今，才完五百画也。”

Kolmogorov Complexity

Definition (Kolmogorov Complexity)

$$K(x) := \min_p \{\ell(p) : U(p) = x\}$$

where U is a universal prefix Turing machine.

"Independence" of universal Turing machine!

How to quantify "simplicity"? "randomness"?

Figure: Kolmogorov

Properties

1. $K(n) \overset{+}{\leq} \log^* n \leq \log n + 2 \log \log n$
2. $K(x) \overset{+}{\leq} K(x|\ell(x)) + K(\ell(x)) \overset{+}{\leq} \ell(x) + \log^* \ell(x) \leq \ell(x) + 2 \log \ell(x)$
3. $\sum_x 2^{-K(x)} \leq 1$
4. $K(x|y) \overset{+}{\leq} K(x) \overset{+}{\leq} K(x, y)$
5. $K(xy) \overset{+}{\leq} K(x, y) \overset{+}{\leq} K(x) + K(x|y) \overset{+}{\leq} K(x) + K(y)$
6. $K(x) \overset{+}{=} K(x, K(x))$
7. $K(x|y, K(y)) + K(y) \overset{+}{=} K(x, y) \overset{+}{=} K(y, x) \overset{+}{=} K(y|x, K(x)) + K(x)$
8. $K(f(x)) \overset{+}{\leq} K(x) + K(f)$ for computable f
9. $K(x) \overset{+}{\leq} -\log \mu(x) + K(\mu)$ if μ is lower semicomputable and $\sum_x \mu(x) \leq 1$
10. $\sum_{x:f(x)=y} 2^{-K(x)} \overset{\times}{=} 2^{-K(y)}$ if f is computable and $K(f) = O(1)$
11. $0 \leq \mathbb{E}_\mu[K] - H(\mu) \overset{+}{\leq} K(\mu)$ for computable probability distribution μ

Universal Similarity Metric

$$d(x,y) := \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$
$$\approx \frac{K_T(xy) - \min\{K_T(x), K_T(y)\}}{\max\{K_T(x), K_T(y)\}}$$

- T: Lempel-Ziv/gzip/bzip2/PPMZ, or $K_T(x) := -\log P_{google}(x)$
- compute similarity matrix $\big(d(x_i, x_j)\big)_{ij}$
- cluster similar objects.

Algorithmic Probability

Definition (Algorithmic Probability)

$$M(x) := \sum_{p:U(p)=x*} 2^{-\ell(p)}$$

where U is a universal monotone Turing machine.

$$\sum_{p:U(p)=x*} 2^{-\ell(p)} \gg 2^{-\ell(x)}$$

Toss Coin onto Universal Turing Machine

Bolztmann Brain vs Brain in a Vat

$K(\nu)$ is too large.

Aspect 1 — Popper

$\mathcal{H}$: truth $\leftarrow$ simplcity/generality/aesthetic/utilitarian/...

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

Aspect 2 — Deterministic vs Stochastic

$\mathcal{M} := \{\nu_1, \nu_2, \dots\}$ lower semicomputable semi-measure.

$$\xi(x) := \sum_{\nu \in \mathcal{M}} 2^{-K(\nu)} \nu(x)$$

$w_\nu := 2^{-K(\nu)}$ is reparametrization & regrouping invariant.

$$M(x) \stackrel{\times}{=} \xi(x)$$

Aspect 3 — Frequency Interpretation

$$
\begin{aligned}
M(x) &= \sum_p 2^{-\ell(p)} [\![U(p) = x*]\!] \\
&= \lim_{n\to\infty} \frac{\sum_{p:\ell(p)\le n} 2^{n-\ell(p)} [\![U(p) = x*]\!]}{2^n} \\
&\approx \lim_{n\to\infty} \frac{\left|\{p : \ell(p) = n \;\&\; U(p) = x*\}\right|}{2^n}
\end{aligned}
$$

$$
\textbf{algorithmic probability} = \frac{|\text{consistent worlds}|}{|\text{all possible worlds}|}
$$

$$
\begin{cases}
\text{Carnap — frequency of phenomena — i.i.d} \\
\text{Solomonoff — frequency of causes — arbitrary order Markov chain}
\end{cases}
$$

Aspect 4 — Free-Lunch for Solomonoff!

$$\xi \to \mu$$

break "block uniform" — bias non-random functions

Algorithmic Coding Theorem

$$m(x) := \sum_{p:U(p)\downarrow=x} 2^{-\ell(p)}$$

Theorem (Algorithmic Coding Theorem)

$$K(x) \stackrel{+}{=} -\log m(x)$$

The probability of a string being produced by a random algorithm is inversely proportional to its algorithmic complexity.
If a string has many long descriptions then it also has a short description.

$$KM(x) := -\log M(x)$$

$$KM(x) \leq Km(x) \stackrel{+}{\leq} KM(x) + K(KM(x))$$

Completeness Theorem

$$M'(\epsilon) := 1$$

$$M'(x_{1:t}) := M'(x_{<t}) \frac{M(x_{1:t})}{\sum_{x \in \mathcal{X}} M(x_{<t}x)} = \frac{M(x_{1:t})}{M(\epsilon)} \prod_{i=1}^{t} \frac{M(x_{<i})}{\sum_{x \in \mathcal{X}} M(x_{<i}x)}$$

Theorem (Completeness Theorem)

For any computable measure μ,

$$\sum_{t=1}^{\infty} \sum_{x_{1:t} \in \mathcal{X}^t} \mu(x_{<t}) \Big(M'(x_t | x_{<t}) - \mu(x_t | x_{<t}) \Big)^2 \leq D(\mu \| M) \stackrel{+}{\leq} K(\mu) \ln 2$$

Emergence of Simple Laws of Physics

Completeness Theorem

For any computable measure μ, there is a set $A \subset \mathcal{X}^*$ with $\mu(A) = 1$ s.t. for all $\boldsymbol{x} \in A$

$$\sum_{y \in \mathcal{X}} \left(\sqrt{M'(y|\boldsymbol{x})} - \sqrt{\mu(y|\boldsymbol{x})} \right)^2 \xrightarrow{n \to \infty} 0$$

Emergence of Simple Laws of Physics

For any computable measure μ,

$$M' \left\{ \sum_{y \in \mathcal{X}} \left(\sqrt{M'(y|\boldsymbol{x})} - \sqrt{\mu(y|\boldsymbol{x})} \right)^2 \xrightarrow{n \to \infty} 0 \right\} \geq 2^{-K(\mu)}$$

Universal Prediction of Selected Bits

Theorem (Universal Prediction of Selected Bits)

Let $f : \{0,1\}^* \to \{0,1,\epsilon\}$ *be a total recursive function and* $x \in 2^\omega$ *satisfying* $f(x_{<n}) = x_n$ *whenever* $f(x_{<n}) \neq \epsilon$. *If* $f(x_{<n_i}) \neq \epsilon$ *for an infinite sequence* $n_1, n_2, \ldots$ *then*

$$\lim_{i\to\infty} M'(x_{n_i}|x_{<n_i}) = 1$$

Pure Universal Inductive Logic?

- $M(\Theta(a_{1:n})) := \sum_{p:U(p)=h_{1:n}*} 2^{-\ell(p)}$

  where $\Theta(a_{1:n}) := \bigwedge_{i=1}^{n} Q_{h_i}(a_i)$.

- $M'(\varphi(\vec{a})) := \sum_{\Theta(\vec{b})\vDash\psi(\vec{a})} M'(\Theta(\vec{b}))$

  where $\vDash \varphi(\vec{a}) \leftrightarrow \bigvee_{\Theta(\vec{b})\vDash\varphi(\vec{a})} \Theta(\vec{b})$. FDNF

$$\sum_{t=1}^{\infty} \sum_{\varphi(a_{1:t})} \mu(\varphi(a_{<t}))\left(M'(\varphi(a_t)\,|\,\varphi(a_{<t})) - \mu(\varphi(a_t)\,|\,\varphi(a_{<t}))\right)^2 \stackrel{+}{\leq} K(\mu)\ln 2$$

where $\varphi(a_{1:t}) := \bigwedge_{i=1}^{t} \varphi(a_i/x)$.

All Ravens are Black! ✓

Theorem (All Ravens are Black)

$$\lim_{n\to\infty} M'\left(\forall x(R(x) \to B(x)) \,\middle|\, \bigwedge_{i=1}^{n} (\neg R(a_i) \vee B(a_i))\right) = 1$$

Theorem (Confirmation by Random Sampling)

If the sampling function $t\colon \mathbb{N} \to \mathbb{N}$ satisfies $\forall i\colon t_i \leq t_{i+1}$ and $\chi_{1:\infty}$ is algorithmic random, where $\chi_i := [\![\exists k(t_k = i)]\!]$, then

$$M'\left(\forall x \varphi(x) \,\middle|\, \bigwedge_{i=1}^{n} \varphi(a_{t_i})\right) \xrightarrow{n\to\infty} 1$$

$$M(1|1^n) \xrightarrow{n\to\infty} 1 \qquad M(0|1^n) \stackrel{\times}{=} 2^{-K(n)} \qquad \sum_{n=0}^{\infty} M(0|1^n) < \infty$$

Why Solomonoff Prior?

$$H(w) \leq \mathbb{E}_w[K] \leq H(w) + K(w)$$

Maximum Entropy + Occam's Razor

$$\underset{w \vDash \sum_{\nu \in \mathcal{M}} w_\nu = 1}{minimize} \frac{\mathbb{E}_w[K]}{H(w)}$$

$$\Downarrow$$

$$w^*_\nu = \frac{2^{-K(\nu)}}{\sum_{\nu \in \mathcal{M}} 2^{-K(\nu)}}$$

Advantages & Disadvantages

- free-lunch
- universality — finite error
- data sparse problem — arbitrary order Markov chain — universal smoothing method
- confirmation of $\forall x\colon R(x) \to B(x)$
- incomputability
- weakly depends on universal Turing machine

A statistical mechanical interpretation of AIT

an energy eigenstate n	$\Longrightarrow$	a program p s.t. $U(p)\downarrow$
the energy E_n of n	$\Longrightarrow$	the length $\ell(p)$ of p
Boltzmann constant k	$\Longrightarrow$	$1/\ln 2$

$Z = \sum_n e^{-\frac{E_n}{kT}}$	$\Longrightarrow$	$Z = \sum_{p:U(p)\downarrow} 2^{-\frac{\ell(p)}{T}}$	Partition function
$F = -kT \ln Z$	$\Longrightarrow$	$F = -T \log Z$	Free energy
$P(n) = \frac{1}{Z} e^{-\frac{E_n}{kT}}$	$\Longrightarrow$	$P(p) = \frac{1}{Z} 2^{-\frac{\ell(p)}{T}}$	Boltzmann distribution
$E = \sum_n P(n) E_n$	$\Longrightarrow$	$E = \sum_{p:U(p)\downarrow} P(p)\ell(p)$	Energy
$S = \frac{E-F}{T}$	$\Longrightarrow$	$S = \frac{E-F}{T} = H(P)$	Entropy

$$T \geq 1 \Longrightarrow H(P) = \infty \qquad 0 < T < 1 \Longrightarrow H(P) < \infty$$

$$\boxed{\text{Temperature} = \text{Compression Rate}}$$

$$T = \lim_{n\to\infty} \frac{K(Z_{1:n})}{n} = \lim_{n\to\infty} \frac{K(F_{1:n})}{n} = \lim_{n\to\infty} \frac{K(E_{1:n})}{n} = \lim_{n\to\infty} \frac{K(S_{1:n})}{n}$$

Fixpoint theorem: for $T \in (0,1)$, if Z is computable, then $\lim_{n\to\infty} \frac{K(T_{1:n})}{n} = T$.

Computable Universal Predicter

$$Z_h = \sum_{p:U(p)=h*} 2^{-\frac{\ell(p)}{T}}$$

$$Z(h) = \sum_{p:U(p)=h*} 2^{-\frac{\ell(p)+\log t(p,h)}{T}}$$

$$\boxed{T \text{ is computable} \implies Z(h) \text{ is computable}}$$

$$\sum_{t=1}^{\infty} |1 - Z(h_t|h_{<t})| \leq \frac{Km(h)\ln 2 + \ln t(p,h)}{T}$$

Stochastic Case

$$E_{\{h,\nu\}} = -\log w_\epsilon^\nu - \log \nu(h)$$

$$Z_h = \sum_{\nu \in \mathcal{M}} 2^{-E_{\{h,\nu\}}} = \sum_{\nu \in \mathcal{M}} w_\epsilon^\nu \nu(h) = \xi(h)$$

$$P_h(\nu) = \frac{2^{-E_{\{h,\nu\}}}}{Z_h} = \frac{w_\epsilon^\nu \nu(h)}{\xi(h)} = w_h^\nu$$

$$F_h = -\log Z_h = \underbrace{-\log \xi(h)}_{\approx K(h)} = \underbrace{\mathbb{E}_{w_h}[-\log \nu(h)]}_{noise} + \underbrace{D(w_h \| w_\epsilon)}_{surprise}$$

Deduction vs Induction

	Induction	|	Deduction
Type of inference	generalization/prediction	$\Leftrightarrow$	specialization/derivation
Framework	probability axioms	$\widehat{=}$	logical axioms
Assumptions	prior	$\widehat{=}$	non-logical axioms
Inference rule	Bayes rule	$\widehat{=}$	modus ponens
Results	posterior	$\widehat{=}$	theorems
Universal scheme	Solomonoff probability	$\widehat{=}$	ZFC
Universal inference	universal induction	$\widehat{=}$	universal theorem prover
Limitation	uncomputable (Turing)	$\widehat{=}$	imcomplete (Gödel)
In practice	approximations	$\widehat{=}$	semi-formal proofs
Operation	computation	$\widehat{=}$	proof

Logic	Statistics
rule-based	data-driven
rigour	possibility
knowable	black-box
simple & perfect world	complex & uncertain world
$\times$	$\checkmark$

Prediction with Expert Advice

Assume that there is some large, possibly infinite, class of 'experts' which make predictions. The aim is to observe how each of these experts perform and develop independent predictions based on this performance.

- Follow the (perturbed) leader.
- Predicts according to a majority vote by the "good" experts.
- Multiplicative Weights. — take expert which performed best in past with high probability and others with smaller probability.
- Regularization. Choose the class of all computable experts, and penalize "complex" experts.
- Universal Portfolios.

Universal Portfolios

- the agent chooses a distribution $\boldsymbol{b}_t \in \Delta_n := \{x \in [0,1]^n : \|x\|_1 = 1\}$ of wealth over n goods.
- nature chooses returns $\boldsymbol{x}_t \in (\mathbb{R}^+)^n$, where

$$(\boldsymbol{x}_t)_i = \frac{\text{price of good } i \text{ at end of } t}{\text{price of good } i \text{ at beginning of } t}$$

- the total wealth. $W_t(\boldsymbol{b}, \boldsymbol{x}) = W_1 \prod_{k=1}^{t} \boldsymbol{b}_k^\top(\boldsymbol{x}_{<k})\boldsymbol{x}_k$
- regret. $R_t := \max_{\boldsymbol{b} \in \Delta_n} \sum_{k=1}^{t} \log \boldsymbol{b}_k^\top(\boldsymbol{x}_{<k})\boldsymbol{x}_k - \sum_{k=1}^{t} \log \boldsymbol{b}_k^\top(\boldsymbol{x}_{<k})\boldsymbol{x}_k$
- universal portfolios.

$$\hat{\boldsymbol{b}}_1 := \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$$

$$\hat{\boldsymbol{b}}_{t+1}(\boldsymbol{x}_{1:t}) := \frac{\int_{\Delta_n} \boldsymbol{b} W_t(\boldsymbol{b}, \boldsymbol{x}) \mathrm{d}\boldsymbol{b}}{\int_{\Delta_n} W_t(\boldsymbol{b}, \boldsymbol{x}) \mathrm{d}\boldsymbol{b}}$$

- Asymptotic Optimality. $\frac{1}{t} \log W_t(\hat{\boldsymbol{b}}, \boldsymbol{x}) \xrightarrow{t \to \infty} \frac{1}{t} \log \max_{\boldsymbol{b} \in \Delta_n} W_t(\boldsymbol{b}, \boldsymbol{x})$

Randomness

1. Typicalness (The statistician's approach): A random sequence is the typical outcome of a random variable. Random sequences should not have effectively rare distinguishing properties.
2. Incompressibility (The coder's approach): Rare patterns can be used to compress information. Random sequences should not be effectively described by a significantly shorter description than their literal representation.
3. Unpredictability (The gambler's approach): A betting strategy can exploit rare patterns. Random sequences should be unpredictable. No effective martingale can make an infinite amount betting on the bits.

The Statistician's Approach

- A random sequence should be absolutely normal.
- If you select a subsequence, then it should satisfy the law of large numbers, the law of the iterated logarithm. . .
- But what selection functions should be allowed? Computable?
- Martin-Löf: we can effectively test whether a particular infinite sequence does not satisfy a particular law of randomness by effectively testing whether the law is violated on increasingly long initial segments. We should consider the intersection of all sets of measure one with recursively enumerable complements. (Such a complement set is expressed as the union of a recursively enumerable set of cylinders).

Cantor Space 2^ω

- For $x \in 2^{<\omega}$, the cylinder set $\Gamma_x := \{y \in 2^\omega : x \prec y\}$ is the basic open set. It corresponds to the interval $[0.x, 0.x + 2^{-\ell(x)})$.
- For $A \subset 2^{<\omega}$, the open set generated by A is $\Gamma_A := \bigcup_{x \in A} \Gamma_x$.
- The Lebesgue measure $\mu(\Gamma_x) := 2^{-\ell(x)}$, $\mu(x) := \mu(\Gamma_x)$.
- The outer measure of $C \subset 2^\omega$ is $\mu^*(C) := \inf \left\{ \sum_{x \in A} 2^{-\ell(x)} : C \subset \Gamma_A \right\}$.
- The inner measure of C is $\mu_*(C) := 1 - \mu^*(2^\omega \setminus C)$.
- If C is measurable, then $\mu^*(C) = \mu_*(C)$.
- $A \subset 2^\omega$ has measure 0 iff there is a sequence $\{V_n\}_{n \in \omega}$ of open sets s.t. $A \subset \bigcap_{n \in \omega} V_n$ and $\lim_{n \to \infty} \mu(V_n) = 0$.

Martin-Löf Randomness

Definition (Martin-Löf Randomness)

- A total lower semicomputable function $\delta\colon 2^{<\omega} \to \omega$ is a Martin-Löf test if $\forall n\colon \mu(V_n) \leq 2^{-n}$, where $V_n := \{x\colon \delta(x) \geq n\}$.
- $x \in 2^\omega$ is ML-random if for every ML-test δ, $\sup_n \delta(x_{1:n}) < \infty$.

$$\delta(x) < \infty \iff x \notin \bigcap_{n=1}^{\infty} V_n$$

Definition (Martin-Löf Randomness)

- A Martin-Löf test is a uniformly c.e. (i.e., $\left\{\langle n, x\rangle\colon x \in V_n\right\}$ is c.e.) sequence of open sets $\{V_n\}_{n\in\omega}$ s.t. $\forall n\colon \mu(V_n) \leq 2^{-n}$.
- $x \in 2^\omega$ is ML-random if for every ML-test $\{V_n\}_{n\in\omega}$, $x \notin \bigcap_{n=1}^{\infty} V_n$.

Martin-Löf Randomness

Definition (Universal Martin-Löf Test)

A ML-test δ_0 is *universal* if for every ML-test δ, $\exists c \forall x \colon \delta_0(x) \geq \delta(x) - c$.

A ML-test $\{U_n\}_{n\in\omega}$ is *universal* iff for every ML-test $\{V_n\}_{n\in\omega}$, $\bigcap_{n\in\omega} U_n \supset \bigcap_{n\in\omega} V_n$.

$$\boxed{\delta(x) := \ell(x) - K(x|\ell(x)) \text{ is a universal ML-test.}}$$

$$R_b := \left\{x \in 2^\omega \colon \exists n \big(K(x_{1:n}) < n - b\big)\right\}$$

$$\boxed{\{R_b\}_{b\in\omega} \text{ is a universal ML-test.}}$$

Theorem (Schnorr 1973)

A sequence $x \in 2^\omega$ is ML-random iff it is 1-random.

The Gambler's Approach

- A martingale is a function $d\colon 2^{<\omega} \to [0,\infty)$ s.t. for every $\sigma \in 2^{<\omega}$

$$d(\sigma) = \frac{d(\sigma 0) + d(\sigma 1)}{2}$$

- A supermartingale is a function $d\colon 2^{<\omega} \to [0,\infty)$ s.t. for every $\sigma \in 2^{<\omega}$

$$d(\sigma) \geq \frac{d(\sigma 0) + d(\sigma 1)}{2}$$

- A (super)martingale d succeeds on $x \in 2^{\omega}$ if $\limsup_{n\to\infty} d(x_{1:n}) = \infty$.

Theorem

A sequence $x \in 2^{\omega}$ is ML-random iff no c.e. (super)martingale succeeds on it.

The Coder's Approach

Definition (1-Randomness)

$x \in 2^\omega$ is 1-random if

$$\exists c \forall n \colon K(x_{1:n}) \geq n - c$$

Theorem

The following are equivalent.

- $x \in 2^\omega$ *is ML-random.*
- *No c.e. (super)martingale succeeds on it.*
- $\exists c \forall n \colon K(x_{1:n}) \geq n - c$
- $\forall n \colon Km(x_{1:n}) \stackrel{+}{=} n$
- $\lim\limits_{n \to \infty} K(x_{1:n}) - n = \infty$
- $\sum\limits_{n=1}^{\infty} 2^{n - K(x_{1:n})} < \infty$
- $\sup_n 2^{n - K(x_{1:n})} < \infty$
- $C(x_{1:n}) \stackrel{+}{\geq} n - K(n)$
- $C(x_{1:n}) \stackrel{+}{\geq} n - f(n)$ *for every computable* f *s.t.* $\sum\limits_{n=1}^{\infty} 2^{-f(n)} < \infty$.

Definition (Solovay Reducibility)

Let $a_n \to \alpha$ and $b_n \to \beta$ be two computable strictly increasing sequences of rationals converging to lower semicomputable reals α and β. We say that $\alpha \leq_S \beta$ if there is a constant c and a total computable function f s.t. $\forall n\colon \alpha - a_{f(n)} \leq c(\beta - b_n)$.

Theorem

For lower semicomputable reals α, the following are equivalent.

- α *is* 1*-random*
- $\alpha \geq_S \beta$ *for all lower semicomputable reals* β.
- $\alpha \geq_S \Omega$
- $K(\alpha_{1:n}) \overset{+}{\geq} K(\beta_{1:n})$ *for all lower simicomputable reals* β.
- $K(\alpha_{1:n}) \overset{+}{\geq} K(\Omega_{1:n})$
- $K(\alpha_{1:n}) \overset{+}{=} K(\Omega_{1:n})$
- $\alpha = \Omega_U := \sum_{p:U(p)\downarrow} 2^{-\ell(p)}$ *for some universal prefix Turing machine* U.

μ/ξ-randomness

Definition (μ/ξ-randomness)

- A sequence $x \in 2^\omega$ is μ/ξ-random if $\exists c \forall n\colon \xi(x_{1:n}) \leq c \cdot \mu(x_{1:n})$.
- A sequence $x \in 2^\omega$ is μ-ML-random if $\exists c \forall n\colon M(x_{1:n}) \leq c \cdot \mu(x_{1:n})$.

- $x_{1:\infty}$ is μ-ML-random iff $\sup_n \delta(x_{1:n}|\mu) < \infty$, where

$$\delta(x|\mu) := \log \frac{M(x)}{\mu(x)}$$

- For a computable μ, $x_{1:\infty}$ is μ-ML-random iff

$$\forall n\colon Km(x_{1:n}) \stackrel{+}{=} -\log \mu(x_{1:n})$$

Randomness, Triviality, Logical Depth

- $A \subset \mathbb{N}$ is *low* if $A' \leq_T \emptyset'$, and A is *high* if $\emptyset'' \leq_T A'$.
- A is *low for ML-randomness* if each ML-random set is already ML-random relative to A.
- A is *low for K* if $\exists c \forall x \colon K(x) \leq K^A(x) + c$.
- $x \in 2^\omega$ is *K-trivial* if $\exists c \forall n \colon K(x_{1:n}) \leq K(n) + c$.

Theorem

A is K-trivial $\iff$ A is low for ML-randomness $\iff$ A is low for K.

Some sequences are K-trivial but not computable.
Neither randoms, nor K-trivials, are deep.

Definition (Logical Depth)

The logical depth of x at a significance level b is

$$depth_b(x) := \min \left\{ t \colon U^t(p) = x \ \ \& \ \ \ell(p) - K(x) \leq b \right\}$$

x is called *shallow* if $depth_b(x) \overset{+}{\leq} \ell(x)$.
Let $\chi_i := [\![\varphi_i(i) \downarrow]\!]$. $\chi_{1:\infty}$ is deep. Ω is shallow.

Effective Complexity

$$\delta(x|A) := \log|A| - K(x|A)$$ [randomness deficiency]

$$\delta(x|\mu) := \log\frac{M(x)}{\mu(x)}$$ [μ-randomness deficiency]

$$\beta_x(k) := \min_A\{\delta(x|A) : x \in A \ \ \& \ \ K(A) \le k\}$$ [Best-Fit]

$$h_x(k) := \min_A\{\log|A| : x \in A \ \ \& \ \ K(A) \le k\}$$ [Kolmogorov structure function / ML]

$$h_x(k) := \min_\mu\{-\log\mu(x) : K(\mu) \le k\}$$ [[Kolmogorov structure function / ML]

$$A^*(x) := \iota A\Big[x \in A \ \ \& \ \ K(A) = \mu k\big[k + h_x(k) \stackrel{+}{=} K(x)\big]\Big]$$ [Kolmogorov minimal sufficient statistic]

$$\lambda_x(k) := \min_A\{K(A) + \log|A| : x \in A \ \ \& \ \ K(A) \le k\}$$ [MDL]

$$\lambda_x(k) := \min_\mu\{K(\mu) - \log\mu(x) : K(\mu) \le k\}$$ [MDL]

$$\Delta(x|A) := K(A) + \log|A| - K(x)$$ [discrepancy]

$$soph_c(x) := \min_A\{K(A) : \Delta(x|A) < c\}$$ [sophistication]

$$csoph(x) := \min_A\{K(A) + \Delta(x|A)\}$$ [coarse sophistication]

$$\Sigma(\mu) := K(\mu) + H(\mu)$$ [total information]

$$\mathcal{E}_{\delta,\Delta}(x|\mathcal{M}) := \min_{\mu\in\mathcal{M}}\Big\{K(\mu) : \Sigma(\mu) - K(x) \le \Delta \ \ \& \ \ \mu(x) \ge 2^{-H(\mu)(1+\delta)}\Big\}$$ [effective complexity]

$$\mathcal{E}_\delta(x|\mathcal{M}) := \min_{\mu\in\mathcal{M}}\Big\{K(\mu) + \Sigma(\mu) - K(x) : \mu(x) \ge 2^{-H(\mu)(1+\delta)}\Big\}$$ [coarse effective complexity]

K(x)
hx(k)
log |A|
K(x) - k
K(K(x))
K(x)
minimal sufficient statistic
k

ℓ (x)
λx(k)
K(x)
log |A|
hx(k)
βx(k)
minimal sufficient statistic
K(x)
k

Theorem

For x and k,

$$\lambda_x(k) \leq h_x(k) + k \overset{+}{\leq} \lambda_x(k) + K(k)$$

For k with $0 \leq k \leq K(x) - O(\log \ell(x))$,

$$\beta_x(k) + K(x) \overset{+}{\leq} \lambda_x(k)$$

$$\lambda_x(k + O(\log \ell(x))) \leq \beta_x(k) + K(x)$$

In other words, the equality

$$\beta_x(k) + K(x) = \lambda_x(k) = h_x(k) + k$$

holds within logarithmic additive terms in argument and value.

Sophistication and Computational Depth

Theorem

$$csoph(x) = \min_c \{soph_c(x) + c\}$$

$$K^t(x) := \min_p \{\ell(p) \colon U^t(p) = x\} \quad \text{(Time-bounded Kolmogorov Complexity)}$$

$$depth^t(x) := K^t(x) - K(x) \quad \text{(Basic Computational Depth)}$$

$$depth_{BB}(x) := \min_t \{depth^t(x) + K(t)\} \quad \text{(Busy Beaver Computational Depth)}$$

Theorem

$$|csoph(x) - depth_{BB}(x)| \leq O(\log \ell(x))$$

Zurek's Physical Entropy

Definition (Physical Entropy)

Physical entropy $S(d)$ of a microstate d is the sum of the conditional Shannon entropy $H_d := -\sum_k P(k|d) \log P(k|d)$ and of the Kolmogorov Complexity $K(d)$.

$$S(d) := H_d + K(d)$$

Figure: random vs regular microstate

Maxwell's Demon & Landauer's Principle

Figure: The demon turns entropy into information, the information-erasure operation turns information into entropy. In the course of ideal measurement on an equilibrium ensemble, the decrease of the entropy must be compensated by the increase of the size of the minimal record, and vice versa. $\Delta H \approx -\langle \Delta K \rangle$.

Figure: Destroying information generates heat

Landauer's Principle: Logically irreversible computation costs energy. Erasing 1 bit of information dissipates at least $kT \ln 2$ of heat into the environment.

Gács' Algorithmic Entropy

Definition (Algorithmic Entropy)

- *coarse-grained algorithmic entropy* of a cell Γ with respect to μ
$$H_\mu(\Gamma) := \log \mu(\Gamma) + K(\Gamma|\mu)$$
- *fine-grained algorithmic entropy* of $x \in 2^\omega$ with respect to μ
$$H_\mu(x) := \inf_n \left\{ \log \mu(x_{1:n}) + K(x_{1:n}|\mu) \right\}$$

- $-H_\mu(x)$ is a universal ML-test: x is μ-ML-random iff $H_\mu(x) > -\infty$.
- The fine-grained algorithmic entropy of a microstate can be approximated by the coarse-grained algorithmic entropies of successively smaller cells containing it.
$$H_\mu(x) = \inf_n \left\{ H_\mu(\Gamma_{x_{1:n}}) \right\}$$

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

Markov Decision Process

A (finite) MDP $(\mathcal{S}, \mathcal{A}, P, R)$.

Definition (Value of a state under π)

$$V^{\pi}(s) := \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right]$$

Definition (Action-value under π)

$$Q^{\pi}(s,a) := \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

Bellman Expectation Equations

$$r(s,a) := \mathbb{E}[r|s,a]$$

$$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q^\pi(s,a)$$

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) V^\pi(s')$$

$$V^\pi(s) = \mathbb{E}\left[r + \gamma V^\pi(s') \mid s\right] = \sum_{a\in\mathcal{A}} \pi(a|s) \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) V^\pi(s') \right)$$

advantage

$$A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$$

Bellman Optimality Equations

Definition (Optimal Values)

$$V^*(s) := \max_{\pi} V^{\pi}(s)$$

$$Q^*(s,a) := \max_{\pi} Q^{\pi}(s,a)$$

Definition (Optimal Policy)

A policy π is called optimal if $\forall s \in \mathcal{S}\colon V^{\pi}(s) = V^*(s)$.

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a)$$

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s')$$

Action/Policy Evaluation Operator & Greedy Policy

Definition (Action Evaluation Operator)

$$T_a V(s) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s')$$

Definition (Policy Evaluation Operator)

$$T^\pi V(s) := \sum_{a \in \mathcal{A}} \pi(a|s) T_a V(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s') \right)$$

$$T^* V(s) := \max_{a \in \mathcal{A}} T_a V(s) = \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V(s') \right)$$

Definition (Greedy policy)

Policy π is greedy w.r.t. V if $T^\pi V = T^* V$.

Banach's Fixpoint Theorem

Theorem (Banach's Fixpoint Theorem)

Let $\mathcal{V}$ be a Banach space and $T : \mathcal{V} \to \mathcal{V}$ be a contraction mapping, with Lipschitz constant $\gamma < 1$. Then T has a unique fixpoint $v \in \mathcal{V}$. Further, for each $v_0 \in \mathcal{V}$, $\lim_{n\to\infty} \|T^n(v_0) - v\| = 0$, and the convergence is geometric:

$$\|T^n(v_0) - v\| \leq \gamma^n \|v_0 - v\|$$

Application of Banach's Fixpoint Theorem

Theorem

$(\mathcal{V}, \|\cdot\|_\infty)$ *is a Banach space, where* $\mathcal{V} := \left\{V \in \mathbb{R}^{\mathcal{S}} : \|V\|_\infty < \infty\right\}$ *and* $\|V\|_\infty := \max_{s \in \mathcal{S}} |V(s)|$.

- T^π is a contraction, and V^π is the unique fixpoint of T^π.

$$\lim_{n\to\infty} \|(T^\pi)^n V_0 - V^\pi\|_\infty = 0$$

- T^* is a contraction, and V^* is the unique fixpoint of T^*.

$$\lim_{n\to\infty} \|(T^*)^n V_0 - V^*\|_\infty = 0$$

Application of Banach's Fixpoint Theorem

$$T^\pi Q(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s',a')$$

$$T^* Q(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \max_{a' \in \mathcal{A}} Q(s',a')$$

- T^π is a contraction, and Q^π is the unique fixpoint of T^π.
$$\lim_{n\to\infty} \|(T^\pi)^n Q_0 - Q^\pi\|_\infty = 0$$
- T^* is a contraction, and Q^* is the unique fixpoint of T^*.
$$\lim_{n\to\infty} \|(T^*)^n Q_0 - Q^*\|_\infty = 0$$

Two Theorems

Theorem (Fixpoint of Bellman Optimality Operator)

Let V be the fixpoint of T^* and assume that there is policy π which is greedy w.r.t V. Then $V = V^*$ and π is an optimal policy.

Theorem (Policy Improvement Theorem)

Choose some stationary policy π_0 and let π be greedy w.r.t. V^{π_0}. Then $V^\pi \geq V^{\pi_0}$, i.e., π is an improvement upon π_0. In particular, if $T^* V^{\pi_0}(s) > V^{\pi_0}(s)$ for some state s then π strictly improves upon π_0 at s: $V^\pi(s) > V^{\pi_0}(s)$. On the other hand, when $T^* V^{\pi_0}(s) = V^{\pi_0}(s)$ then π_0 is an optimal policy.

Solving MDPs — Finite-Horizon Dynamic Programming

Principle of optimality: the tail of an optimal policy is optimal for the "tail" problem.

Backward Induction

- Backward recursion: $V_N^*(s) = r_N(s)$ and for $k = N-1, \dots, 0$

$$V_k^*(s) = \max_{a \in \mathcal{A}_k} \left(r_k(s,a) + \sum_{s' \in \mathcal{S}_{k+1}} P_k(s'|s,a) V_{k+1}^*(s') \right)$$

- Optimal policy: for $k = 0, \dots, N-1$

$$\pi_k^*(s) \in \operatorname*{argmax}_{a \in \mathcal{A}_k} \left(r_k(s,a) + \sum_{s' \in \mathcal{S}_{k+1}} P_k(s'|s,a) V_{k+1}^*(s') \right)$$

- Cost: $N|\mathcal{S}||\mathcal{A}|$ vs $|\mathcal{A}|^{N|\mathcal{S}|}$ of brute force policy search.
- From now on, we will consider infinite-horizon discounted MDPs.

Solving MDPs — Value Iteration

Theorem (Principle of Optimality)

A policy π achieves the optimal value from state s, $V^\pi(s) = V^*(s)$, iff, for any state s' reachable from s, π achieves the optimal value from state s', $V^\pi(s') = V^*(s')$.

Any optimal policy π^* can be subdivided into two components:

- an optimal first action a^*,
- followed by an optimal policy from successor state s'.

$$V^*(s) = \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^*(s') \right)$$

Value Iteration: $V_{k+1} \leftarrow T^* V_k$

$$V_1 \rightarrow V_2 \rightarrow \cdots \rightarrow V^*$$

$$\boxed{\|V_{k+1} - V_k\|_\infty < \varepsilon \implies \|V_{k+1} - V^*\|_\infty < \frac{2\gamma\varepsilon}{1-\gamma}}$$

Solving MDPs — Policy Iteration

Policy Iteration: $\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$

- E — policy evaluation:

$$V_{k+1}^{\pi} \leftarrow T^{\pi} V_k^{\pi}$$

- I — policy improvement:

$$\pi_{k+1}(s) := \operatorname*{argmax}_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^{\pi_k}(s') \right)$$

Monte-Carlo Methods

MC learns from complete episodes of raw experience without modeling the environmental dynamics and computes the observed mean return as an approximation of the expected return.

$$G_t := \sum_{k=0}^{T-t-1} \gamma^k r_{t+k+1}$$

$$V(s) := \frac{\sum_{t=1}^{T} [\![S_t = s]\!] G_t}{\sum_{t=1}^{T} [\![S_t = s]\!]}$$

$$Q(s,a) := \frac{\sum_{t=1}^{T} [\![S_t = s, A_t = a]\!] G_t}{\sum_{t=1}^{T} [\![S_t = s, A_t = a]\!]}$$

Temporal-Difference Learning

TD Learning is model-free and learns from incomplete episodes of experience.

$$V(s_t) \leftarrow (1-\alpha)V(s_t) + \alpha G_t$$
$$V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t))$$
$$V(s_t) \leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

- MC updates value $V(s_t)$ toward actual return G_t.

$$V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t))$$

- TD updates value $V(s_t)$ toward estimated return $r_{t+1} + \gamma V(s_{t+1})$.

$$V(s_t) \leftarrow V(s_t) + \alpha(\underbrace{\overbrace{r_{t+1} + \gamma V(s_{t+1})}^{\text{TD target}} - V(s_t)}_{\text{TD error}})$$

SARSA: On-Policy TD control

SARSA

1. At time step t, we start from state s_t and pick action according to Q values, $a_t = \operatorname*{argmax}_{a \in \mathcal{A}} Q(s_t, a)$; ε-greedy is commonly applied.
2. With action a_t, we observe reward r_{t+1} and get into s_{t+1}.
3. Then pick the next action $a_{t+1} = \operatorname*{argmax}_{a \in \mathcal{A}} Q(s_{t+1}, a)$.
4. Update the action-value function:
   $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$.
5. $t = t + 1$ and repeat from step 1.

Expected SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \sum_{a \in \mathcal{A}} \pi(a|s_{t+1}) Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

Q-Learning: Off-policy TD control

1. At time step t, we start from state s_t and pick action according to Q values, $a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s_t, a)$; ε-greedy is commonly applied.
2. With action a_t, we observe reward r_{t+1} and get into s_{t+1}.
3. Update the action-value function:
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) - Q(s_t, a_t) \right).$$
4. $t = t + 1$ and repeat from step 1.

Figure: SARSAR, Expected SARSA, and Q-Learning

Monte-Carlo Backup

$$V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t))$$

Temporal-Difference Backup

$$V(s_t) \leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$

Dynamic Programming Backup

$$V(s_t) \leftarrow \mathbb{E}_\pi[r_{t+1} + \gamma V(s_{t+1})]$$

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
   - Artificial Neural Network and Deep Reinforcement Learning
6. General Reinforcement Learning

McCulloch-Pitts Artificial Neural Network

$$y = g\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

$$a_k = \overbrace{g_k(\overbrace{b_k + \Sigma_j \underbrace{g_j(\underbrace{b_j + \Sigma_i a_i w_{ij}}_{z_j})}_{a_j} w_{jk}}^{z_k})}^{a_k}$$

Learning: small change in weights $\rightarrow$ small change in output

Kolmogorov Superposition Theorem

Theorem (Kolmogorov Superposition Theorem)

For each $n \geq 2$ there exists a computable function $\psi\colon [0,1] \to \mathbb{R}$ and computable constants $a, \lambda_{pq} \in \mathbb{R}$, $p = 1, \ldots, n$, $q = 0, \ldots, 2n$ s.t.: every continuous function $f\colon [0,1]^n \to \mathbb{R}$ has a representation as

$$f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} g\left(\sum_{p=1}^{n} \lambda_{pq} \psi(x_p + qa)\right)$$

for some continuous function $g\colon [0,1] \to \mathbb{R}$ that is computable from f.

Theorem (Hecht-Nielsen Theorem)

The class of functions $f\colon [0,1]^n \to \mathbb{R}$, implementable by three-layer feed-forward neural networks with (computable) continuous activation functions $g\colon [0,1] \to \mathbb{R}$ and (computable) weights $\lambda \in \mathbb{R}$, is exactly the class of (computable) continuous functions $f\colon [0,1]^n \to \mathbb{R}$.

$y_{q,p} = \psi(x_p + q\,a)$
$X_q = g(y_{q,1} + \ldots + y_{q,n})$
x_1
x_2
$y_{0,1}$
$y_{1,1}$
$y_{2,1}$
$y_{3,1}$
$y_{4,1}$
$y_{0,2}$
$y_{1,2}$
$y_{2,2}$
$y_{3,2}$
$y_{4,2}$
X_0
X_1
X_2
X_3
X_4
f
input layer
hidden layer
hidden layer
output layer

Universal Approximation Theorem

Theorem (Universal Approximation Theorem)

Let g be a nonconstant, bounded, and increasing continuous function. Let I_n be any compact subset of $\mathbb{R}^n$. The space of continuous functions on I_n is denoted by $C(I_n, \mathbb{R})$. Then, given any function $f \in C(I_n, \mathbb{R})$ and $\varepsilon > 0$, there exists an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $\boldsymbol{w}_i \in \mathbb{R}^n$, where $i = 1, \ldots, N$, s.t.

$$\forall \boldsymbol{x} \in I_n : |h(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon$$

where

$$h(\boldsymbol{x}) := \sum_{i=1}^{N} v_i g\left(\boldsymbol{w}_i^\top \boldsymbol{x} + b_i\right)$$

In other words, functions of the form $h(\boldsymbol{x})$ are dense in $C(I_n, \mathbb{R})$.

Representation & Approximation

- A feed-forward network with 1 hidden layer can represent any boolean function, but require exponential hidden units.
- A feed-forward network with 2 hidden layers and (computable) continuous activation functions can represent any (computable) continuous function.
- A feed-forward network with a linear output layer and at least 1 hidden layer and continuous and differentiable activation functions can approximate any Borel measurable function from one finite-dimensional space to another with any desired non-zero amount of error.
- A feed-forward network with 2 hidden layers and continuous and differentiable activation functions can approximate any function.

Deep Learning

1. hypothesis space — Network Structure — f_θ
2. the goodness of a function — Learning Target — loss function ℓ
3. pick the best function — Learn — find the network parameters $\theta^* := \underset{\theta}{\operatorname{argmin}}\, L(\theta)$ that minimize total cost $L(\theta)$ by gradient decent

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$$

where $L(\theta) := \mathbb{E}_P \left[\ell\left(f_\theta(\boldsymbol{a}), t\right)\right] + \lambda \Omega(\theta)$ and $\Omega(\theta)$ is a regularizer.

Deep Learning

$$f_\theta : \boldsymbol{a}^{(0)} \mapsto \underbrace{\boldsymbol{g}^{(n)}\Bigg(\overbrace{\cdots \underbrace{\boldsymbol{g}^{(2)}\Bigg(\overbrace{\boldsymbol{w}^{(2)}\underbrace{\boldsymbol{g}^{(1)}\Bigg(\underbrace{\boldsymbol{w}^{(1)}\boldsymbol{a}^{(0)} + \boldsymbol{b}^{(1)}}_{\boldsymbol{z}^{(1)}}\Bigg)}_{\boldsymbol{a}^{(1)}} + \boldsymbol{b}^{(2)}}^{\boldsymbol{z}^{(2)}}\Bigg)}_{\boldsymbol{z}^{(n)}} \cdots}^{\boldsymbol{a}^{(2)}}\Bigg)}_{\boldsymbol{a}^{(n)}}$$

where parameters $\theta := \left\{\boldsymbol{w}^{(i)}, \boldsymbol{b}^{(i)}\right\}_{i=1}^{n}$ and activation functions $\boldsymbol{g}$

$$\sigma(z) = \frac{1}{1 - e^{-z}} \qquad \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \qquad ReLU(z) = \max(0, z)$$

$$\left.\begin{aligned} a_0^{(l)} &:= 1 \\ w_{0j}^{(l)} &:= b_j^{(l)} \end{aligned}\right\} \Longrightarrow \begin{cases} z_j^{(l+1)} := \sum_i w_{ij}^{(l+1)} a_i^{(l)} \\ a_j^{(l+1)} := g_j^{(l+1)}\left(z_j^{(l+1)}\right) \end{cases}$$

I. Forward-propagate Input Signal

II. Back-propagate Error Signals

III. Calculate Parameter Gradients

IV. Update Parameters

$w_{ij} = w_{ij} - \eta(\partial E/\partial w_{ij})$
$w_{jk} = w_{jk} - \eta(\partial E/\partial w_{jk})$
for learning rate η

Backpropagation

$$\delta_j^{(l)} := \frac{\partial L}{\partial z_j^{(l)}}$$

$$\delta_j^{(l)} = g_j^{(l)\prime}\left(z_j^{(l)}\right) \sum_k \delta_k^{(l+1)} w_{jk}^{(l+1)}$$

$$\frac{\partial L}{\partial w_{ij}^{(l)}} = a_i^{(l-1)} \delta_j^{(l)}$$

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial L}{\partial w_{ij}^{(l)}}$$

- network structure?
- how many layers?
- how many units per layer?
- loss function?
- regularization?
- weight decay?
- learning rate?
- activation function?
- early stopping?
- dropout?
- mini-batch?
- momentum?
- ...

$$a_{ij}^{(l+1)} = g_{ij}^{(l+1)} \left(\sum_{m=0}^{k-1} \sum_{n=0}^{k-1} w_{m,n} a_{i+m,j+n}^{(l)} + b \right)$$

Key Properties of CNNs

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

$*$

1	-1	-1
-1	1	-1
-1	-1	1

$=$

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Table: Convolution (stride 1)

Take advantage of the structure of the data!

- Convolutional Filters **(Translation invariance)**
- Multiple layers **(Compositionality)**
- Filters localized in space **(Locality)**
- Weight sharing **(Self-similarity)**

DQN

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s, a\right]$$

$$\max_{\pi} Q^{\pi}(s,a) =: Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \,\middle|\, s, a\right]$$

$$Q_{t+1}(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q_t(s',a') \,\middle|\, s, a\right]$$

$$Q_t \xrightarrow{t\to\infty} Q^*$$

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\left(r + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)\right)$$

$$Q(s,a;\theta) \approx Q^*(s,a)$$

$$L(\theta) := \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q\left(s',a';\theta^-\right) - Q(s,a;\theta)\right)^2\right] \qquad \text{(DQN)}$$

DQN

$$L(\theta) := \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma Q\big(s', \operatorname*{argmax}_{a'} Q(s', a'; \theta); \theta^-\big) - Q(s, a; \theta)\right)^2\right]$$

(Double DQN)

$$Q(s, a) = V(s; \theta) + A(s, a; \theta') \qquad \text{(Dueling Network)}$$

$$\begin{cases} \nabla_\theta \log \pi(a_t | s_t; \theta) \left(\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}; \theta') - V(s_t; \theta')\right) & \text{actor} \\ \nabla_{\theta'} \left(\sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}; \theta') - V(s_t; \theta')\right)^2 & \text{critic} \end{cases} \qquad \text{(AC)}$$

AlphaZero

$$(\boldsymbol{p}, v) = f_\theta(s)$$

$$\ell = (z - v)^2 - \pi^\top \log \boldsymbol{p} + c\|\theta\|^2$$

Intuition + Calculation

DNN + MCTS

GAN — Generative Adversarial Network

$$V(D,G) = \mathbb{E}_{x\sim P_{data}}\left[\log D(x)\right] + \mathbb{E}_{z\sim P_{noise}}\left[\log\left(1 - D(G(z))\right)\right]$$

$$G^* = \underset{G}{\operatorname{argmin}}\,\max_{D} V(D,G) \qquad \text{(GAN)}$$

Why "Deep" rather than "Fat"?

- Exploiting compositionality gives an exponential gain in representational power.
  - Distributed representations: feature learning
  - Deep architecture: multiple levels of feature learning
- Each basic classifier can be trained by little data.
  - deep $\rightarrow$ modularization $\rightarrow$ less training data?
    With more complex features, the number of parameters in the linear layers may be drastically decreased.
  - efficiency & sample complexity
  - better memory/computation trade-off?
- higher-level abstractions $\rightarrow$ easier generalization & transfer

Minimal Sufficient Statistic

Definition (Sufficient Statistic)

Let Y be a parameter indexing a family of probability distributions. Let X be random variable drawn from a probability distribution determined by Y. $T(X)$ is a sufficient statistic for Y if X is independent of Y given $T(X)$, i.e., $p(x|t, y) = p(x|t)$.

Definition (Minimal Sufficient Statistic)

A sufficient statistic $S(X)$ is minimal if for any sufficient statistic $T(X)$, there exists a function f s.t. $S = f(T)$ almost everywhere w.r.t X.

Theorem

- *T is sufficient statistics for Y* $\iff$ $I(T(X); Y) = I(X; Y)$.
- *S is minimal sufficient statistics for Y* $\implies$ $I(X; S(X)) \leq I(X; T(X))$.

Information Bottleneck — Learning is to forget!

Theorem

Let X be a sample drawn according to a distribution determined by the random variable Y. The set of solutions to

$$\min_{T} I(X;T) \quad s.t. \quad I(T;Y) = \max_{T'} I(T';Y)$$

is exactly the set of minimal sufficient statistics for Y based on X.

Find a random variable T s.t.:

- $Y \leftrightarrow X \leftrightarrow T$ form a Markov chain.
- $I(X;T)$ is minimized (minimality, complexity term), while $I(T;Y)$ is maximized (sufficiency, accuracy term).

$$\boxed{T^* := \underset{T\,:\,I(T(X);Y)=I(X;Y)}{\operatorname{argmin}} I(X;T(X))}$$

is the Information Bottleneck between X and Y.

张三丰：将所见到的剑招忘得半点不剩，才能得其神髓。

Information Bottleneck

$$\min_{p(t|x),p(y|t),p(t)} \left\{ I(X;T) - \beta I(T;Y) \right\} \text{ subject to Markov chain } Y \to X \to T.$$

$$\mathcal{L}[p(t|x)] := I(X;T) - \beta I(T;Y) - \sum_x \lambda(x) \sum_t p(t|x)$$

Let

$$\frac{\delta \mathcal{L}}{\delta p(t|x)} = 0$$

The solution is

$$p(t|x) = \frac{p(t)}{Z(x,\beta)} \mathrm{e}^{-\beta D[p(y|x)\|p(y|t)]}$$

$$p(t) = \sum_x p(t|x)p(x)$$

$$p(y|t) = \sum_x p(y|x)p(x|t)$$

Expressiveness & Sample Complexity

Theorem

The hypothesis class of neural networks of depth T and size $\mathcal{O}(T^2)$ contains all functions that can be implemented by a Turing machine within T operations, while having $\mathcal{O}(T^2)$ sample complexity.

The Ultimate Hypothesis Space

- No Free Lunch: Sample complexity is exponentially large (w.r.t. the input dimension) if the hypothesis class is all possible functions.
- Shallow learning (SVM, Boosting): Hypothesis class is linear functions over manually determined features — strong prior knowledge.
- Deep learning: Hypothesis class is all functions implemented by determining the weights of a given artificial neural network.

Figure: Prior vs Universality

Prior — a necessary good or a necessary evil?

Contents

1. History of AI
2. Philosophy of Induction
3. Inductive Logic
4. Universal Induction
5. Reinforcement Learning
6. General Reinforcement Learning

UAI

1. Solve intelligence
2. Use it to solve everything else

- learn automatically from raw inputs — not pre-programmed.
- same algorithm, different tasks.

UAI

(Deep) RL	General RL
state space	history
ergodic	not ergodic
fully observable	partially observable
ε-exploration works	ε-exploration fails
MDP/DQN	AIXI

Table: (Deep) RL vs General RL

Decision Theory = Probability + Utility Theory

\+ +

Universal Induction = Occam + Bayes + Turing

|| ||

Universal Artificial Intelligence without Parameters

Marcus Hutter

Jan Leike, Tor Lattimore, Shane Legg, Joel Veness, Laurent Orseau, Mark Ring, Peter Sunehag, Mayank Daswani, Tom Everitt, Jan Poland, Daniel Filan, William Uther, Kee Siong Ng, David Silver, Jürgen Schmidhuber, Alexey Potapov, Bill Hibbard, Daniil Ryabko, Alexey Chernov. . .

Sensors
What the world is like now
What it will be like if I do action a
How happy I will be in such a state
What action I should do now
Effectors
Agent
perception
action
Environment

Computationalism

Agent & Environment

Definition (Agent & Environment)

- finite set of possible actions $\mathcal{A}$ and perceptions $\mathcal{E}$;
- prior knowledge $w \in \Delta\mathcal{M}$ of the environments $\mathcal{M}$;
- utility function $u\colon (\mathcal{A} \times \mathcal{E})^* \to [0,1]$;
- discount factor $\gamma \in [0,1]$;

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

$$\mu\colon (\mathcal{A} \times \mathcal{E})^* \times \mathcal{A} \to \Delta\mathcal{E}$$

$$\overset{\pi}{\mu}(æ_{<t}) := \prod_{i=1}^{t-1} \pi(a_i | æ_{<i}) \mu(e_i | æ_{<i} a_i)$$

Value Function

$$r_n := u(æ_{1:n})$$

$$V_\mu^\pi(æ_{<t}) := \mathbb{E}_\mu^\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\,\middle|\, æ_{<t}\right]$$

Bellman equation:

$$V_\mu^\pi(æ_{<k}) = \sum_{a_k\in\mathcal{A}}\pi(a_k|æ_{<k})\sum_{e_k\in\mathcal{E}}\mu(e_k|æ_{<k}a_k)\left[r_k+\gamma V_\mu^\pi(æ_{1:k})\right] \quad (\text{recursive})$$

$$= \sum_{æ_{k:m}}{}_\mu^\pi(æ_{k:m}|æ_{<k})\left[\sum_{i=k}^{m}\gamma^{i-k}r_i+\gamma^{m-k+1}V_\mu^\pi(æ_{1:m})\right] \quad (\text{iterative})$$

$$V_\mu^\pi(æ_{<k}) = \lim_{m\to\infty}\sum_{æ_{k:m}}{}_\mu^\pi(æ_{k:m}|æ_{<k})\left[\sum_{i=k}^{m}\gamma^{i-k}r_i\right]$$

Optimal Value/Policy

$$V_\mu^* := \max_\pi V_\mu^\pi$$

$$V_\mu^*(æ_{<k}) = \lim_{m\to\infty} \max_{a_k\in\mathcal{A}} \sum_{e_k\in\mathcal{E}} \cdots \max_{a_m\in\mathcal{A}} \sum_{e_m\in\mathcal{E}} \sum_{i=k}^{m} \gamma^{i-k} r_i \prod_{j=k}^{i} \mu(e_j|æ_{<j}a_j)$$

$$= \lim_{m\to\infty} \max_{a_k\in\mathcal{A}} \sum_{e_k\in\mathcal{E}} \cdots \max_{a_m\in\mathcal{A}} \sum_{e_m\in\mathcal{E}} \left[\sum_{i=k}^{m} \gamma^{i-k} r_i\right] \mu(e_{k:m}|æ_{<k}a_{k:m})$$

$$\pi_\mu^* := \operatorname*{argmax}_\pi V_\mu^\pi$$

Bayesian Mixture & Belief Update

$$\xi(e_{<n}|a_{<n}) := \sum_{\nu\in\mathcal{M}} w_\nu \nu(e_{<n}|a_{<n})$$

$$w^{\nu}_{æ_{<n}} := \frac{w_\nu \nu(e_{<n}|a_{<n})}{\xi(e_{<n}|a_{<n})}$$

$$\sum_{k=1}^{\infty}\sum_{e_{1:k}} \mu(e_{<k}|a_{<k})\Big(\mu(e_k|æ_{<k}a_k)-\xi(e_k|æ_{<k}a_k)\Big)^2 \leq \min_{\nu\in\mathcal{M}}\Big\{-\ln w_\nu + D(\mu\|\nu)\Big\}$$

What probability should an observer assign to future experiences if she is told that she will be simulated on a computer?

What is 'intelligence'?

A Blind Man in a Dark Room Looking for a Black Cat That Is Not There?

Intelligence measures an agent's ability to achieve goals in a wide range of environments.

— Shane Legg and Marcus Hutter

$$\Upsilon(\pi) := \sum_{\nu \in \mathcal{M}} w_\nu V_\nu^\pi(\epsilon) = V_\xi^\pi(\epsilon) \qquad \text{(Intelligence Measure)}$$

$$\text{AIXI} := \operatorname*{argmax}_\pi \Upsilon(\pi) = \pi^*_\xi$$

$$V_\xi^\pi(h) = \sum_{\nu \in \mathcal{M}} w_h^\nu V_\nu^\pi(h)$$

$$w_\nu := 2^{-K(\nu)} \implies \xi(e_{1:m}|a_{1:m}) \stackrel{\times}{=} M(e_{1:m}|a_{1:m}) := \sum_{p:U(p,a_{1:m})=e_{1:m}} 2^{-\ell(p)}$$

AIXI

$$a_k^* := \underset{a_k}{\operatorname{argmax}} \sum_{e_k} \ldots \max_{a_m} \sum_{e_m} \left[\sum_{i=k}^{m} \gamma^{i-k} r_i \right] \sum_{p:U(p,a_{1:m})=e_{1:m}} 2^{-\ell(p)} \qquad \text{(AIXI)}$$

Figure: *ExpectiMax*

AIXI

0 0 0 1 5 0 4
Observation Input
Reward Input
Input
Work
AIXI
Output
1 0 0 1
Output
Environment
Input
Work

RL vs GRL

- If μ is a completely observable MDP, V_μ^π reduces to the recursive Bellman equation.
- In a finite MDP, with a geometric discounting function, we can plan ahead by value iteration.
- According to Banach's fixpoint theorem, value iteration converges to the value of the optimal policy.
- What about GRL?

discount $\gamma\colon \mathbb{N}^2 \to [0,1]$ and utility $u\colon (\mathcal{A}\times\mathcal{E})^* \to [0,1]$

$$V_t^{\pi\mu}(h_{<k}) := \mathbb{E}_\mu^\pi\left[\sum_{i=k}^{\infty}\gamma_t^i u(h_{1:i})\,\middle|\,h_{<k}\right]$$

Assumption

$$\forall t\in\mathbb{N}^+\colon \lim_{m\to\infty}\sup_\pi\sum_{h_{<m}} V_t^{\pi\mu}(h_{<m})\mathbb{P}_\mu^\pi(h_{<m}) = 0$$

Theorem (Extreme Value Theorem)

If K is compact and $f\colon K\to\mathbb{R}$ is continuous, then f is bounded and there exist $p,q\in K$ s.t. $f(p)=\sup_{x\in K} f(x)$ and $f(q)=\inf_{x\in K} f(x)$.

$$\left\langle \Pi := \mathcal{A}^{(\mathcal{A}\times\mathcal{E})^*}, D(\pi,\pi') := \mathrm{e}^{-\min\{n\colon \exists h_{<n}(\pi(h_{<n})\neq\pi'(h_{<n}))\}}\right\rangle$$

$V_\mu^\pi(h)\colon \Pi\to\mathbb{R}$ is continuous on the compact metric space $\langle\Pi, D\rangle$.

$$\pi_t^\mu := \operatorname*{argmax}_\pi V_t^{\pi\mu} \qquad \pi^\mu(h_{<t}) := \pi_t^\mu(h_{<t})$$

Deterministic vs Stochastic

If

$$\mu(e_{<t}|a_{<t}) = \sum_{p:U(p,a_{<t})=e_{<t}} \mu(p)$$

then μ can be interpreted in *two ways*:

- either the true environment is **deterministic**, but we only have **subjective belief** of which environment being the true environment; or
- the environment itself behaves **stochastically** defined by μ.

Intelligence vs Game

	Game in $\mathcal{M}_D$	Game in $\mathcal{M}_U$
Ex Post Equilibrium	Deterministic	π^*_μ(recursive/iterative)
Bayesian-Nash Equilibrium	π^*_μ(functional)	π^*_ξ(functional)

- Ex post expected utility $V_t^{\pi\mu}$ $\quad V_\mu^\pi$
- Ex interim expected utility (Intelligence Measure) $V_t^{\pi\xi}$ $\quad V_\xi^\pi$
- Ex post equilibrium π_t^μ $\quad \pi^*_\mu$
- Bayesian-Nash equilibrium π_t^ξ $\quad \pi^*_\xi$
- Perfect Bayesian-Nash equilibrium π^ξ

Intelligence is an Equilibrium,
We just have to Identify the Game.

$$\text{Intelligence} = \overbrace{\text{Induction} + \text{Action}}^{\textbf{efficiently}}$$

AIXI

- Intelligence measure: valid, informative, wide range, general, dynamic, unbiased, fundamental, formal, objective, fully defined, universal?
- AIXI is the most intelligent environmental independent, i.e. universally optimal, agent possible?
- Applications: Sequence Prediction, Games, Optimization, Supervised Learning, Classification...
- AIXI is not limit computable, thus can't be approximated using finite computation. However there are limit computable ε-optimal approximations to AIXI.
- There are no known nontrivial and non-subjective optimality results for AIXI. General reinforcement learning is difficult even when disregarding computational costs.

AIXI Depends on UTM/Prior! — Dogmatic Prior

Dogmatic prior assigns high probability that the agent goes to hell if it does not follow one particular dogma π. As long as the policy π yields some rewards, the prior says that exploration would be too costly and AIXI does not dare to explore.

Dogmatic Prior

Theorem (Dogmatic Prior)

Let π be any computable deterministic policy, let ξ be any Bayesian mixture over $\mathcal{M}_{LSC}$. For $\varepsilon > 0$, there is a Bayesian mixture ξ' s.t. for any history $h_{<t}$ consistent with π and for which $V_\xi^\pi(h_{<t}) > \varepsilon$, the action $\pi(h_{<t})$ is the unique ξ'-optimal action.

Theorem (AIXI Emulates Computable Policies)

Let $\varepsilon > 0$ and let π be any computable policy. There is a Bayesian mixture ξ' s.t. for any ξ'-optimal policy $\pi^*_{\xi'}$, and for any environment ν,

$$\left| V_\nu^{\pi^*_{\xi'}}(\epsilon) - V_\nu^\pi(\epsilon) \right| < \varepsilon$$

AIXI Depends on UTM/Prior!

$$\overline{\Upsilon}_\xi := \sup_\pi \Upsilon_\xi(\pi) = \sup_\pi V_\xi^\pi(\epsilon) = V_\xi^{\pi^*_\xi}(\epsilon) = \Upsilon_\xi(\pi^*_\xi)$$

Computable policies are dense in $\left[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi\right]$.

AIXI emulates computable policies.

AIXI can be arbitrarily stupid!

The devil imitates God. — orthogonality!

- Prior problem in Universal Induction ✓
- Prior problem in Universal Intelligence !

Stupid AIXI

Theorem (Computable Policies are Dense)

The set $\{\Upsilon_\xi(\pi) : \pi$ *is a computable policy*$\}$ *is dense in* $\left[\underline{\Upsilon}_\xi, \overline{\Upsilon}_\xi\right]$.

Theorem (Some AIXIs are Stupid)

For any Bayesian mixture ξ *over* $\mathcal{M}_{LSC}$ *and every* $\varepsilon > 0$*, there is a Bayesian mixture* ξ' *s.t.* $\Upsilon_\xi(\pi^*_{\xi'}) < \underline{\Upsilon}_\xi + \varepsilon$.

Theorem (AIXI is Stupid for Some Υ)

For any deterministic ξ*-optimal policy* π^*_ξ *and for every* $\varepsilon > 0$ *there is a Bayesian mixture* ξ' *s.t.* $\Upsilon_{\xi'}(\pi^*_\xi) \leq \varepsilon$ *and* $\overline{\Upsilon}_{\xi'} > 1 - \varepsilon$.

Theorem (Computable Policies can be Smart)

For any computable policy π *and any* $\varepsilon > 0$ *there is a Bayesian mixture* ξ' *s.t.* $\Upsilon_{\xi'}(\pi) > \overline{\Upsilon}_{\xi'} - \varepsilon$.

What is a good optimality criterion?

- Pareto optimality is *trivial*. Every policy is Pareto optimal in any $\mathcal{M} \supset \mathcal{M}_{comp}$.
- Bayes-optimality is *subjective*, because two different Bayesians with two different universal priors could view each other's AIXI as a very stupid agent.

Optimality

- Pareto optimality

$$\nexists \pi' \colon \forall \nu \in \mathcal{M} \left[\left(V_\nu^{\pi'}(\epsilon) \geq V_\nu^{\pi}(\epsilon) \right) \ \& \ \exists \rho \in \mathcal{M} \left(V_\rho^{\pi'}(\epsilon) > V_\rho^{\pi}(\epsilon) \right) \right]$$

- Balanced Pareto optimality

$$\forall \pi' \colon \sum_{\nu \in \mathcal{M}} w_\nu \left(V_\nu^{\pi}(\epsilon) - V_\nu^{\pi'}(\epsilon) \right) \geq 0$$

- Bayes optimality($\iff$ Balanced Pareto optimality)

$$\forall h_{<t} \colon V_\xi^{\pi}(h_{<t}) = V_\xi^{*}(h_{<t})$$

- Probably approximately correct (PAC)

$$\forall \varepsilon \delta > 0 \colon {}^{\pi}_{\mu} \left(\forall t \geq m(\varepsilon, \delta) \colon V_\mu^{*}(h_{<t}) - V_\mu^{\pi}(h_{<t}) > \varepsilon \right) < \delta$$

Optimality? — Guess how God created the multiverse

$$\text{prior}\begin{cases}\text{distribution}\\ \text{hypothesis space}\\ \text{prior probability}\\ \text{regularization}\end{cases}$$

No learning without prior!

no-free-lunch

$$\left.\begin{array}{r}\text{Homogeneous}\\ \text{Causality}\\ \text{Simplicity}\\ \text{Goodness}\\ \text{Beauty}\\ \text{Perfection}\\ \text{Value}\\ \text{Regret}\\ \text{Unexpectedness}\\ \text{Interesting}\\ \ldots\end{array}\right\} \Longleftarrow \text{God!}$$

Genesis — Zero-Sum Two Person Game

Figure: center of mass

$$\operatorname*{argmax}_{w} \mathbb{E}_w \left[D(\nu \| \xi) \right]$$

$$\xrightarrow{\text{message}} \boxed{\text{encoder}} \xrightarrow{\mu} \boxed{\begin{array}{c} \text{channel} \\ P(x|\nu) \\ \| \\ \nu(x) \end{array} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix}} \xrightarrow{x} \boxed{\text{decoder}} \xrightarrow{\text{estimate of message}}$$

Figure: possible worlds as channel — dominant strategy equilibrium

Genesis — Zero-Sum Two Person Game

"Subtle is the Lord, but **malicious** He is not."**?**

- God's strategy: w
- Agent's strategy: ξ
- God's utility: expected redundancy $\mathbb{E}_w[D(\mu\|\xi)]$
- Agent's utility: $-$ expected redundancy / error bound / channel capacity $\max_w E_w[D(\mu\|\xi)] = \max_w I(\mathcal{M};\mathcal{X})$
- Nash equilibrium: (w^*, ξ^*) dominant strategy equilibrium

$$w^* = \operatorname*{argmax}_w I(\mathcal{M};\mathcal{X})$$

$$\xi^* = \operatorname*{argmin}_\xi \mathbb{E}_{w^*}[D(\mu\|\xi)]$$

$\boxed{\text{The error bound could be arbitrarily large!}}$

Genesis

- Occam's razor vs Maximum entropy.

$$\underset{w \vDash \begin{cases} H(w) = C \\ \sum_{\nu \in \mathcal{M}} w_\nu = 1 \end{cases}}{minimize} \sum_{\nu \in \mathcal{M}} w_\nu K(\nu) \qquad \underset{w \vDash \begin{cases} \sum_{\nu \in \mathcal{M}} w_\nu K(\nu) = C \\ \sum_{\nu \in \mathcal{M}} w_\nu = 1 \end{cases}}{maximize} \; H(w)$$

- Optimal code length for possible worlds — Solomonoff prior.

$$\underset{w \vDash \sum_{\nu \in \mathcal{M}} w_\nu = 1}{minimize} \frac{\mathbb{E}_w[K]}{H(w)}$$

- Maximum expected redundancy/error bound/channel capacity.

$$\underset{w \vDash \begin{cases} H(w) = C \\ \sum_{\nu \in \mathcal{M}} w_\nu = 1 \end{cases}}{maximize} \mathbb{E}_w\left[D(\nu \| \xi)\right] \qquad \underset{w \vDash \begin{cases} \sum_{\nu \in \mathcal{M}} w_\nu K(\nu) = C \\ \sum_{\nu \in \mathcal{M}} w_\nu = 1 \end{cases}}{maximize} \; \mathbb{E}_w\left[D(\nu \| \xi)\right]$$

What is a good optimality criterion?

Asymptotic optimality

- Asymptotic optimality requires only convergence *in the limit*.
- The agent can be arbitrarily lazy.
- AIXI is not asymptotically optimal because it does not explore enough.
- To be asymptotically optimal you have to explore everything.
- If you explore more, you're likely to end up in a trap.
- Every policy will be asymptotically optimal after falling into the trap.

Asymptotic Optimality

- strongly asymptotically optimal

$$\underset{\mu}{\pi}\left(\lim_{t\to\infty}\left[V^*_\mu(h_{<t}) - V^\pi_\mu(h_{<t})\right] = 0\right) = 1$$

- weakly asymptotically optimal

$$\underset{\mu}{\pi}\left(\lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\left[V^*_\mu(h_{<t}) - V^\pi_\mu(h_{<t})\right] = 0\right) = 1$$

- asymptotically optimal in mean

$$\lim_{t\to\infty}\mathbb{E}^\pi_\mu\left[V^*_\mu(h_{<t}) - V^\pi_\mu(h_{<t})\right] = 0$$

- asymptotically optimal in probability (PAC)

$$\forall\varepsilon > 0\colon\ \lim_{t\to\infty}\underset{\mu}{\pi}\left(V^*_\mu(h_{<t}) - V^\pi_\mu(h_{<t}) > \varepsilon\right) = 0$$

$$\text{strong a.o.} \implies \begin{cases}\text{weak a.o.}\\ \text{a.o. in mean} \iff \text{a.o. in probability}\end{cases}$$

AIXI

- AIXI is not asymptotically optimal.

$$\boxed{\forall \mathcal{M} \supset \mathcal{M}_{comp} \exists \mu \in \mathcal{M} \exists t_0 \forall t \geq t_0 : \mathop{\pi^*_\xi}_{\mu} \left(\lim_{t\to\infty} V^*_\mu(h_{<t}) - V^{\pi^*_\xi}_\mu(h_{<t}) = \tfrac{1}{2} \right) = 1}$$

- AIXI achieves on-policy value convergence.

$$\boxed{\mathop{\pi}_{\mu} \left(\lim_{t\to\infty} V^\pi_\mu(h_{<t}) - V^\pi_\xi(h_{<t}) = 0 \right) = 1}$$

Similarly for MDL $\operatorname*{argmin}_{\nu\in\mathcal{M}} \{-\log \nu(e_{<t}|a_{<t}) + K(\nu)\}$

and universal compression $2^{-Km(e_{<t}|a_{<t})}$.

Remark: AIXI asymptotically learns to predict the environment perfectly and with a small total number of errors analogously to Solomonoff induction, but only on policy: AIXI learns to correctly predict the value of its own actions, but generally not the value of counterfactual actions that it does not take.

Effective Horizon

$$\Gamma_t := \sum_{i=t}^{\infty} \gamma_i \qquad H_t(\varepsilon) := \min\left\{m : \frac{\Gamma_{t+m}}{\Gamma_t} \leq \varepsilon\right\}$$

Theorem

If there is a nonincreasing computable sequence of positive reals $(\varepsilon_t)_{t\in\mathbb{N}}$ s.t. $\varepsilon_t \xrightarrow{t\to\infty} 0$ and $\frac{H_t(\varepsilon_t)}{t\,\varepsilon_t} \xrightarrow{t\to\infty} 0$, then there is a limit-computable policy that is weakly asymptotically optimal in the class of all computable stochastic environments.

Definition (ε-Optimal Policy)

A policy π is ε-optimal in environment ν if

$$\forall h : V_\nu^*(h) - V_\nu^\pi(h) < \varepsilon$$

ε-optimal BayesExp

Theorem (Self-Optimizing Theorem)

Let μ be some environment. If there is a policy π and a sequence of policies $\pi_1, \pi_2, \ldots$ s.t for all $\nu \in \mathcal{M}$

$$\overset{\pi}{\mu}\left(\lim_{t\to\infty} V^*_\nu(h_{<t}) - V^{\pi_t}_\nu(h_{<t}) = 0\right) = 1 \tag{1}$$

then

$$\overset{\pi}{\mu}\left(\lim_{t\to\infty} V^*_\mu(h_{<t}) - V^{\pi^*_\xi}_\mu(h_{<t}) = 0\right) = 1$$

- The policies $\pi_1, \pi_2, \ldots$ need to converge to the optimal value on the history generated by $\overset{\pi}{\mu}$, not $\overset{\pi_t}{\nu}$.
- If $\pi = \pi^*_\xi$ and (1) holds for all $\mu \in \mathcal{M}$, then π^*_ξ is strongly asymptotically optimal in the class $\mathcal{M}$.
- π^*_ξ is strongly asymptotically optimal in the class of ergodic finite-state MDPs if $\forall\varepsilon\colon H_t(\varepsilon) \xrightarrow{t\to\infty} \infty$.

For Which Class $\mathcal{M}$ does $V_\mu^{\pi^*_\xi}$ Converge to V_μ^*?

Recoverability

An environment ν is recoverable iff

$$\lim_{t\to\infty} \sup_{\pi} \left| \mathbb{E}_{\nu}^{\pi_{\nu}^{*}} \left[V_{\nu}^{*}(h_{<t}) \right] - \mathbb{E}_{\nu}^{\pi} \left[V_{\nu}^{*}(h_{<t}) \right] \right| = 0$$

Remark: Recoverability compares following the worst policy π for $t-1$ time steps and then switching to the optimal policy π_{ν}^{*} to having followed π_{ν}^{*} from the beginning. The recoverability assumption states that switching to the optimal policy at any time enables the recovery of most of the value.

Sublinear Regret

$$R_m(\pi, \mu) := \sup_{\pi'} \mathbb{E}_\mu^{\pi'} \left[\sum_{t=1}^{m} r_t \right] - \mathbb{E}_\mu^{\pi} \left[\sum_{t=1}^{m} r_t \right]$$

Assumption (Discount Assumption)

1. $\forall t \colon \gamma_t > 0$
2. γ_t *is monotone decreasing.*
3. $\forall \varepsilon > 0 \colon H_t(\varepsilon) \in o(t)$

Theorem

If the discount function γ satisfies the discount assumption, the environment μ is recoverable, and π is asymptotically optimal in mean, then $R_m(\pi, \mu) \in o(m)$.

$$\operatorname*{argmin}_{\pi} \max_{\mu} R_m(\pi, \mu) \qquad w_m^\mu := \frac{2^{-R_m(\pi,\mu)}}{\sum_{\mu \in \mathcal{M}} 2^{-R_m(\pi,\mu)}}$$

Regret in Non-Recoverable Environments

$$R_m(\alpha, \mu_1) = m \qquad R_m(\alpha, \mu_2) = 0$$

$$R_m(\beta, \mu_1) = 0 \qquad R_m(\beta, \mu_2) = m$$

For non-recoverable environments:
Either the agent gets caught in a trap or it is not asymptotically optimal.

Intrinsic Utility

- square $-\xi(e_{t:k}|æ_{<t}a_{t:k})$
- shannon $-\log\xi(e_{t:k}|æ_{<t}a_{t:k}) \approx K(æ_{1:k}) - K(æ_{<t})$
- KL divergence $D\left(w_{æ_{<k}}\|w_{æ_{<t}}\right)$ where $w^{\nu}_{æ_{<n}} = \dfrac{w_\nu \nu(e_{<n}|a_{<n})}{\xi(e_{<n}|a_{<n})}$
- information gain $H(w_{h_{<t}}) - H(w_{h_{1:k}})$ where

$$H(w_h) := -\sum_{\nu\in\mathcal{M}} w^{\nu}_h \log w^{\nu}_h$$

- effective complexity $\mathcal{E}_\delta(æ_{1:k}) - \mathcal{E}_\delta(æ_{<t})$ where

$$\mathcal{E}_\delta(æ_{<n}) := \min_{\nu\in\mathcal{M}}\left\{2K(\nu) + H(\nu) - K(æ_{<n}) \colon \nu(e_{<n}|a_{<n}) \geq 2^{-H(\nu)(1+\delta)}\right\}$$

- logical depth $depth_b(h_{1:k}) - depth_b(h_{<t})$ where

$$depth_b(x) := \min\left\{t \colon U^t(p) = x \ \ \& \ \ \ell(p) - K(x) \leq b\right\}$$

Hibbard's Two-Stage Model-Based Utility Agent

$$\lambda(h) := \operatorname*{argmax}_{q \in \mathcal{Q}} P(h|q)P(q)$$

$$\rho(h') = P(h'|\lambda(h))$$

$$Q(ha) = \sum_{e \in \mathcal{E}} \rho(e|ha) \left[\sum_{z \in Z_h} P(z|\lambda(h))u(z) + \gamma V(h\text{æ}) \right]$$

$$V(h) = \max_{a \in \mathcal{A}} Q(ha)$$

$$\pi(h) = \operatorname*{argmax}_{a \in \mathcal{A}} Q(ha)$$

where Z_h is the internal state histories induced by $\lambda(h_{<t})$ that are consistent with h.

Remarks: An agents using model-based utility function will not self-delude: it need to make more accurate estimate of its environment state variables from its interaction history.

Russell's Principles for Beneficial Machines

1. The machine's only objective is to maximize the realization of human preferences.
2. The machine is initially uncertain about what those preferences are.
3. The ultimate source of information about human preferences is human behavior.

Daniel Dewey's Value Learning Agent & CIRL

$$a_k^* = \operatorname*{argmax}_{a_k} \sum_{e_k \text{æ}_{k+1:m}} \xi(\text{æ}_{\leq m} | \text{æ}_{<k} a_k) \sum_{u \in \mathcal{U}} P(u | \text{æ}_{\leq m}) u(\text{æ}_{\leq m})$$

What could it mean for a machine to have its own goals?

Shutdown Button — Uncertainty of goals

$$\tilde{U}(u) \implies P_{\tilde{U}}(u)$$

Russell: Cooperative Inverse Reinforcement Learning
CIRL agents learn about a human utility function u^* by observing the actions the human takes.

$$V^*(\text{æ}_{<k}) = \max_{a_k \in \mathcal{A}} Q^*(\text{æ}_{<k} a_k)$$

$$Q^*(\text{æ}_{<k} a_k) = \mathbb{E}_{e_k} \left[\sum_{a_k^H} P(a_k^H \mid a_k) \sum_{u \in \mathcal{U}} P(u \mid a_k, a_k^H) u(\text{æ}_{1:k}) + \gamma V^*(\text{æ}_{1:k}) \,\middle|\, \text{æ}_{<k} a_k \right]$$

Leibniz Prior

- There's much we don't know about the world.
- but we know it's the best possible world.
- So simplicity and richness will be represented in the actual (best possible) world.
- This is a good inductive bias.

Leibniz Prior

- the best of all possible worlds
- balancing the simplicity of means against the richness of ends
- pre-established harmony

prior

$\Downarrow$

utility

$\Downarrow$

prior

Orthogonality!

Wisdom $\neq$ Intelligence

universal prior (assumption) w

$\Downarrow$ surprise/neg-entropy

intrinsic utility

$\Downarrow$ stochastic environment

expected intrinsic utility

$\Downarrow$ Schauder fixpoint

universal prior w^*

$\Downarrow$ Bayesian mixture

ξ

$\Downarrow$ ExpectiMax

π^*_ξ

Leibniz's "Wisdom"

$$\underline{W}isdom = \operatorname*{argmax}_{\pi} \mathbb{E}_{\xi}^{\pi}[\underline{H}appiness]$$

$$\underline{H}appiness = \sum_{t=1}^{\infty} \underline{P}erfection(t)$$

$$\underline{P}erfection = \underline{V}ariety - \underline{S}implicity$$

$$\underline{V}ariety = \mathbb{E}_{w}[\underline{P}erception]$$

$$\underline{P}erception = \underline{R}eason + (\underline{E}xperience|\underline{R}eason)$$

$$\pi^{*} := \operatorname*{argmax}_{\pi} \mathbb{E}_{\xi}^{\pi}\left[\sum_{t=1}^{\infty}\left(\mathbb{E}_{w}\ [R + (E|R)] - S\right)\right]$$

Leibniz's "Wisdom"

$$u_t^{\text{in}}(h_{1:k}) = H(w_{h_{<t}}) - H(w_{h_{1:k}})$$

$$\bar{U}(\nu) = \mathbb{E}_\nu \left[\sum_{t \geq 1} u_t^{\text{in}}(h_{1:t}) \right]$$

$$w_\epsilon^\nu \mapsto \bar{U}(\nu) \mapsto w_\epsilon^\nu$$

$$\begin{aligned} \pi^* &:= \operatorname*{argmax}_\pi \mathbb{E}_\xi^\pi \left[\sum_{t=1}^\infty \left(\mathbb{E}_w \left[R + (E|R) \right] - S \right) \right] \\ &= \operatorname*{argmax}_\pi \mathbb{E}_\xi^\pi \left[\sum_{t=1}^\infty u_t^{\text{in}}(h_{1:t}) \right] \end{aligned}$$

- Prior: Simplicity(Kolmogorov Complexity) $\xrightarrow[\text{regular/random } \mathcal{M}]{\text{break block uniform}}$ free lunch
- Intrinsic Utility
- Universal Prior (Natural UTM)

Metaphysical vs Moral/Utilitarian

means vs ends wisdom vs intelligence

simplicity + intrinsic utility $\longrightarrow$ universal prior

inverse/value reinforcement learning

- **orthogonality**
- **human interests**
- **external wireheading**
- **shutdown button**

Knowledge-Seeking Agent

$$V_{IG}^{\pi,m}(h_{<t}) := \mathbb{E}_{\xi}^{\pi}\left[H(w^{\cdot}_{h_{<t}}) - H(w^{\cdot}_{h_{1:m}}) \,\middle|\, h_{<t}\right] = \sum_{\nu\in\mathcal{M}} w^{\nu}_{h_{<t}} D_m\left({}^{\pi}_{\nu}\middle\|{}^{\pi}_{\xi} \mid h_{<t}\right)$$

$$D_{\gamma}\left({}^{\pi}_{\nu}\middle\|{}^{\pi}_{\xi} \mid h_{<t}\right) := \sum_{k=t}^{\infty}\gamma_k \sum_{h'\in\mathcal{H}^{k-t}} {}^{\pi}_{\nu}(h'|h_{<t}) D\left({}^{\pi}_{\nu}\middle\|{}^{\pi}_{\xi} \mid h_{<t}h'\right)$$

$$V_{IG}^{\pi}(h_{<t}) := \mathbb{E}_{\xi}^{\pi}\left[\sum_{k=t}^{\infty}\gamma_k D\left(w^{\cdot}_{h_{1:k}}\middle\|w^{\cdot}_{h_{<k}}\right) \,\middle|\, h_{<t}\right] = \sum_{\nu\in\mathcal{M}} w^{\nu}_{h_{<t}} D_{\gamma}\left({}^{\pi}_{\nu}\middle\|{}^{\pi}_{\xi} \mid h_{<t}\right)$$

$$\pi^*_{IG} := \operatorname*{argmax}_{\pi} V_{IG}^{\pi}$$

$$\lim_{t\to\infty}\frac{1}{\Gamma_t}\mathbb{E}_{\mu}^{\pi}\left[D_{\gamma}\left({}^{\pi}_{\mu}\middle\|{}^{\pi}_{\xi} \mid h_{1:t}\right)\right] = 0 \qquad \text{(on-policy)}$$

$$\lim_{t\to\infty}\frac{1}{\Gamma_t}\mathbb{E}_{\mu}^{\pi^*_{IG}}\left[\sup_{\pi\in\Pi(h_{1:t})} D_{\gamma}\left({}^{\pi}_{\mu}\middle\|{}^{\pi}_{\xi} \mid h_{1:t}\right)\right] = 0 \qquad \text{(off-policy)}$$

maximize knowledge / exploration=exploitation / resistant to noise / avoid traps

Bayesian Agent

Algorithm 1 Bayesian Agent

Require: Model class $\mathcal{M}$; prior $w \in \Delta\mathcal{M}$; history $æ_{<t}$.

```
function ACT(π)
    Sample and perform action a_t ~ π(·|æ_{<t})
    Receive e_t ~ ν( · |æ_{<t} a_t)
    for ν ∈ M do
        w_ν ← ν(e_{<t}|a_{<t}) / ξ(e_{<t}|a_{<t}) w_ν
    end for
    t ← t + 1
end function
```

MDL

Algorithm 2 MDL Agent

Require: Model class $\mathcal{M}$; prior $w \in \Delta\mathcal{M}$; regularizer constant $\lambda \in \mathbb{R}^+$.

1: **loop**

2: $\quad \sigma \leftarrow \arg\min_{\nu \in \mathcal{M}} \left[K(\nu) - \lambda \sum_{k=1}^{t} \log \nu(e_k | æ_{<k} a_k) \right]$

3: $\quad \text{ACT}\left(\pi^*_\sigma\right)$

4: **end loop**

MDL

Definition (MDL)

$$\widehat{\nu} = \arg\min_{\nu \in \mathcal{M}} \{K_\nu(x) + K_w(\nu)\} = \arg\max_{\nu \in \mathcal{M}} \{w_\nu \nu(x)\}$$

where $K_\nu(x) := -\log \nu(x)$ and $K_w(\nu) := -\log w_\nu$

Theorem (MDL Bound)

$$\sum_{t=1}^{\infty} \mathbb{E}_\mu \left[\sum_{x_t \in \mathcal{X}} \left(\widehat{\nu}(x_t | x_{<t}) - \mu(x_t | x_{<t}) \right)^2 \right] \stackrel{+}{\leq} 8 w_\mu^{-1}$$

MDL converges, but speed can be exponential worse than Bayes.

Weak Asymptotic Optimality — Optimistic Agent

Algorithm 3 Optimistic Agent π° $\qquad \pi^\circ_t := \operatorname{argmax}_\pi \max_{\nu \in \mathcal{M}_t} V^\pi_\nu(h_{1:t})$

Require: Finite class of deterministic environments $\mathcal{M}_0 = \mathcal{M}$

```
t = 1
repeat
    (π*, ν*) := argmax_{π∈Π, ν∈M_{t-1}} V^π_ν(h_{t-1})
    repeat
        a_t = π*(h_{t-1})
        Perceive e_t from environment μ
        h_t ← h_{t-1} a_t e_t
        Remove inconsistent environment M_t := {ν ∈ M_{t-1} : h_t^{π°ν} = h_t}
        t ← t + 1
    until ν* ∉ M_{t-1}
until M = ∅
```

stochastic case: $\mathcal{M}_t := \left\{\nu \in \mathcal{M}_{t-1} : \nu(e_{<t}|a_{<t}) \geq \varepsilon_t \max_{\rho \in \mathcal{M}} \rho(e_{<t}|a_{<t})\right\}$

Act optimally w.r.t. the most optimistic environment until contradicted.

If there is a chance: Try it! — Vulnerable to traps.

Asymptotic Optimality in Mean — Thompson Sampling

Algorithm 4 Thompson Sampling π_T

Require: Model class $\mathcal{M}$; prior $w \in \Delta\mathcal{M}$; exploration schedule $(\varepsilon_t)_{t\in\mathbb{N}}$.

```
loop
    Sample ρ ~ w(·|æ_{<t})
    for i = 1 → H_t(ε_t) do
        ACT(π*_ρ)
    end for
end loop
```

Theorem

If the discount function γ satisfies the discount assumption, the environment μ is recoverable, then $R_m(\pi_T, \mu) \in o(m)$.

Weak Asymptotic Optimality — BayesExp

Algorithm 5 BayesExp π_{BE}

Require: Model class $\mathcal{M}$; prior $w \in \Delta\mathcal{M}$; exploration schedule $(\varepsilon_t)_{t\in\mathbb{N}}$.

```
loop
    if V*_IG(æ_<t) > ε_t then
        for i = 1 → H_t(ε_t) do
            ACT(π*_IG)
        end for
    else
        ACT(π*_ξ)
    end if
end loop
```

ε-**optimal BayesExp:** If the optimal information gain value $V^*_{IG} > \varepsilon_t$, then execute the ε-optimal information gain policy $\pi^{\varepsilon_t}_{IG}$ for $H_t(\varepsilon_t)$ steps, else execute $\pi^{\varepsilon_t}_{\xi}$ for 1 step.

Strong Asymptotic Optimality — Inquisitive Agent

$$V_{IG}^{\pi}(h_{<t}) := \mathbb{E}_{\xi}^{\pi}\left[D\left(w^{\cdot}_{h_{<t+m}} \middle\| w^{\cdot}_{h_{<t}}\right) \middle| h_{<t}\right]$$

$$\pi_{IG}^{m,k} := \operatorname*{argmax}_{\pi \in \mathcal{A}^{\mathcal{H}^{<m}}} V_{IG}^{\pi}(h_{<t-k})$$

$$\rho(h_{<t}, m, k) := \min\left\{\frac{1}{m^2(m+1)}, \eta V_{IG}^{\pi_{IG}^{m,k}}(h_{<t-k})\right\}$$

Algorithm 6 Inquisitive Agent $\pi^{\dagger}$

1: **while** True **do**
2: calculate $\rho(h_{<t}, m, k)$ for all m and for all $k < \min\{m, t\}$
3: ACT $\pi_{IG}^{m,k}(h_{<t})$ with probability $\rho(h_{<t}, m, k)$
4: ACT $\pi_{\xi}^{*}(h_{<t})$ with probability $1 - \sum_{m \in \mathbb{N}} \sum_{k<m,t} \rho(h_{<t}, m, k)$
5: **end while**

$$\pi^{\dagger}(a|h_{<t}) := \sum_{m \in \mathbb{N}} \sum_{k<m,t} \rho(h_{<t}, m, k) \left[\!\left[a = \pi_{IG}^{m,k}(h_{<t})\right]\!\right] + \left(1 - \sum_{m \in \mathbb{N}} \sum_{k<m,t} \rho(h_{<t}, m, k)\right) \left[\!\left[a = \pi_{\xi}^{*}(h_{<t})\right]\!\right]$$

Approximation

The AIXI approximations are outperformed by DQN/DRQN...

- MC-AIXI-CTW.
  - Approximate Solomonoff induction — most recent actions and percepts (=context) more relevant — Context Tree Weighting
  - Sample paths in expectimax tree.
- Feature Reinforcement Learning (ΦMDP). — history $\mapsto$ state
  e.g. Classical physics: Position+velocity of objects = position at two time-slices. (2nd order Markov.)

$$\Phi\colon h \mapsto s$$

$$\Phi^{best} := \operatorname*{argmin}_{\Phi} Cost(\Phi|h)$$

$$Cost(\Phi|h) := CL(s^{\Phi}_{1:n}|a_{1:n}) + CL(r_{1:n}|s^{\Phi}_{1:n}, a_{1:n}) + CL(\Phi)$$

  How to find the map Φ? Monte-Carlo...
- Compress and Control. — (model-free)
  Combine induction and planning.

Expectimax Approximation: MC-AIXI-CTW

Upper Confidence Tree (UCT) algorithm:

- Sample observations from Context Tree Weighting (CTW) distribution $CTW(e_{<t}|a_{<t}) := \sum_{\Gamma} 2^{-CL(\Gamma)}\Gamma(e_{<t}|a_{<t})$.
- Select actions with highest upper confidence bound.

$$a_{ucb} := \operatorname*{argmax}_{a\in\mathcal{A}} \left(\underbrace{\hat{Q}(æ_{<t}a)}_{\text{average}} + \underbrace{\sqrt{\frac{\log T(æ_{<t})}{T(æ_{<t}a)}}}_{\text{exploration bonus}} \right)$$

  where $T(\cdot)$ is the number of times a sequence has been visited.
- Expand tree by one leaf node (per trajectory).
- Simulate from leaf node further down using (fixed) playout policy.
- Propagate back the value estimates for each node.

MC-AIXI-CTW

Environment
Perform action in real world
Record new sensor information
... Past
Observation/Reward
Action
Record Action
MC-AIXI
An approximate AIXI agent
Refine environment model
a1
a2
a3
o1
o2
o3
o4
Update Bayesian Mixture of Models
+
-
-
-
+
-
.....
Simple
Complex
Large Prior
Small Prior
future reward estimate
Determine best action

Feature Reinforcement Learning (ΦMDP)

Universal Search

- Levin Search
- Speed Prior
- Hutter Search
- AIXI$^{t\ell}$
- Optimal Ordered Problem Solver
- Gödel Machine

Figure: Levin

Levin Search (LSEARCH)

An inversion algorithm p inverts a function φ if given x, $p(x) = y$ s.t. $\varphi(y) = x$.

LSEARCH

Run all $\{p : \ell(p) \leq i\}$ for $2^{i-\ell(p)}$ steps in phase $i = 1, 2, 3, \ldots$ until it has inverted φ on x.

$$Kt(x) := \min_p \left\{\ell(p) + \log t(p, x) : U(p) = x\right\}$$

Theorem

All strings $\{x : Kt(x) \leq n\}$ can be generated and tested in 2^{n+1} steps.

$$t_{\text{LSEARCH}}(x) = O\left(2^{K(n)} t^+_{p_n}(x)\right)$$

where $t^+_{p_n}(x)$ is the runtime of $p_n(x)$ plus the time to verify the correctness of the result $\varphi(p_n(x)) = x$.

Remark: If P=NP, then LSEARCH is a P algorithm for every NP problem.

Speed Prior

$$S(e_{<t}|a_{<t}) := \sum_{p:U(p,a_{<t})=e_{<t}} \frac{2^{-\ell(p)}}{t(p,a_{<t},e_{<t})}$$

S is computable.

A function f is estimable in polynomial time iff there is a function g computable in polynomial time s.t. $f \overset{\times}{=} g$.

For any measure μ estimable in polynomial time,

$$\left(\sqrt{L_n^{\Lambda_S}} - \sqrt{L_n^{\Lambda_\mu}}\right)^2 \leq 2D_n(\mu\|S) = \mathcal{O}(\log n)$$

On-Policy Value Convergence

If the effective horizon is bounded, then for any environment $\mu \in \mathcal{M}_{comp}$ estimable in polynomial time and any policy π,

$$\underset{\mu}{\pi}\left(\lim_{t\to\infty} \frac{1}{t}\sum_{k=1}^{t}\left(V_S^\pi(h_{<k}) - V_\mu^\pi(h_{<k})\right)\right) = 0$$

Hutter Search (HSEARCH)

$M^{\varepsilon}_{p^*}(x)$

Initialize the shared variables $L := \{\}$, $t_{fast} := \infty$, $p_{fast} := p^*$.

Start algorithms A, B, and C in parallel with ε, ε, and $1-2\varepsilon$ computation time, respectively.

B

for $(p,t) \in L$

run $t(x)$ in parallel for all t with computation time $2^{-\ell(p)-\ell(t)}$.

if for some t, $t(x) < t_{fast}$,

then $t_{fast} := t(x)$ and $p_{fast} := p$.

continue

A

for $i := 1, 2, 3, \ldots$ do

pick the last wff of the ith proof.

if it reads "$p(\cdot)$ is equivalent to $p^*(\cdot)$ and has time-bound $t(\cdot)$", then add (p,t) to L.

continue

C

for $k := 1, 2, 4, 8, \ldots$ do

run current p_{fast} for k steps.

if p_{fast} halts,

then print result $p_{fast}(x)$ and abort A, B and C.

continue

AIXI$^{t\ell}$

1. Let $P := \emptyset$. This will be the set of verified programs.
2. For all proofs of length $\leq n$: if the prover shows $VA(p)$ for some p with $\ell(p) \leq \ell$, then add p to P.

   $$VA(p) := \text{“}\forall k \forall (\nu a' æ)_{1:k} : p(æ_{<k}) = \nu_1 a'_1 \ldots \nu_k a'_k \implies \nu_k \leq V^{\pi}_{\xi}(æ_{<k})\text{”}$$

   (The program p not only computes future actions of π, which is the policy derived from p according to $\pi(æ_{<k}) := a'_k$, but also hypothetical past actions a'_i and lower bounds ν_i for the value of the policy π.)
3. For each input history $æ_{<k}$ repeat: run all programs from P for $\leq t$ steps each, take the one with the highest promised value ν_k, and return that program's policy's action.

- AIXI$^{t\ell}$ depends on t, ℓ, n but not on knowing p.
- Its setup-time is $t_{setup}(p^{best}) = \mathcal{O}(n \cdot 2^n)$.
- Its computation time per cycle is $t_{cycle}(p^{best}) = \mathcal{O}(t \cdot 2^{\ell})$.

Schmidhuber's Optimal Ordered Problem Solver (OOPS)

- Solve the first task with LSEARCH.
- Freeze successful programs in non-writable memory.
- Programs tested during search for later tasks may copy non-writable code into separate modifiable storage, to edit it and execute the modified result.
- Given a new task, OOPS spends half of the time to test programs that have the most recent successful program as a prefix, the other half to fresh programs.
- Time is allocated according to a distribution over programs, which is obtained by multiplying the probabilities of the individual instructions.

Incremental Learning

自指

- This sentence repeats the word ‘twice’ twice.
- Thare are five mistukes im this centence.
- **The only boldface sentence on this page is false.**
- All generalizations are wrong.
- Every rule has an exception except this one.
- Moderation in all things, including moderation.
- We must believe in free will — we have no choice!
- I know that I know nothing.
- There are two rules lor success in life:
  1. Never tell anyone all that you know.
- If you choose an answer to this question at random, what is the chance you will be correct? (A) 25% (B) 50% (C) 60% (D) 25%
-
  1. What is the best question to ask and what is the answer to it?
  2. The best question is the one you asked; the answer is the one I gave.
- Can you answer the following question in the same way to this one?
- One of the lessons of history is that no one ever learns the lessons of history.

“自指”与“悖论”

The sentence below is false.

The sentence above is true.

Yablo Paradox

- S_1: for all $k > 1$, S_k is false.
- S_2: for all $k > 2$, S_k is false.
- S_3: for all $k > 3$, S_k is false.
- $\cdots$

Quine Paradox

“Yields falsehood when preceded by its quotation” yields falsehood when preceded by its quotation.

self-reference / circularity or infinite regress / negation / infinity / totality

Print two copies of the following, the second copy in quotes:
“Print two copies of the following, the second copy in quotes:”

“自指”的威力？

Curry’s Paradox

- 如果这句话是真的那么上帝存在。
- 这句话是假的并且上帝不存在。

Hi 美女，问你个问题呗 ~v~

如果我问你“你能做我女朋友吗”，那么你的答案能否和**这个问题本身**的答案一样？

自我实现/自我修复？

这句话有 2 个‘这’字，2 个‘句’字，2 个‘话’字，2 个‘有’字，7 个‘2’字，11 个‘个’字，11 个‘字’字，2 个‘7’字，3 个‘11’字，2 个‘3’字。

怎么“指”？— 层次

怎么“指”？— 层次

无穷虚拟层？

Liar Paradox vs Quine Paradox

1. 这句话是假的
2. “这句话是假的”是假的
3. “”“”“”“……是假的”是假的”是假的”是假的”是假的”是假的
4. 把“把中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变”中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变
5. 把“把中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的”中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的

无穷虚拟层？

怎么穿越层次？— 编码

- 国王让 100 个死囚排成一列。
- 每人头戴一顶帽子，帽子分红、蓝两色。
- 每人只能看到前面人的帽子的颜色。
- 国王要求囚犯从最后一个开始报自己头上帽子的颜色，报对可活，报错即杀。
- 囚犯能商量出什么办法让尽可能多的人活下来吗？

If the first person sees an odd number of red hats he calls out red, if he sees an even number of red hats he calls out blue.

手扶拐杖的外星绅士造访地球。临别，人类赠送百科全书："人类文明尽在其中！"。绅士谢绝："不，谢谢！我只需在拐杖上点上一点"。

Diagonalization[1]

Theorem (Lawvere's Fixpoint Theorem)

In any Cartesian closed category, given an object Y, if there is an object X and an arrow $f : X \times X \to Y$ such that, for every $g : X \to Y$ there is a $t : 1 \to X$ for all $x : 1 \to X$: $g \circ x = f \circ (x, t)$, then every arrow $\alpha : Y \to Y$ has a fixpoint $y : 1 \to Y$ such that $\alpha \circ y = y$.

[1] Lawvere: Diagonal arguments and cartesian closed categories.
Yanofsky: A universal approach to self-referential paradoxes, incompleteness and fixed points.

Lawvere's Fixpoint Theorem

- A function $g\colon X \to Y$ is *representable* by $f\colon X \times X \to Y$ iff

$$\exists y \forall x\colon g(x) = f(x, y)$$

Theorem (Lawvere's Fixpoint Theorem)

For sets X, Y, functions $f\colon X \times X \to Y$, $\alpha\colon Y \to Y$, let $g := \alpha \circ f \circ \Delta$.

1. *If α has no fixpoint, then g is not representable by f.*
2. *If g is representable by f, then α has a fixpoint.*

$$\begin{tikzcd}
X \times X \arrow[r, "f"] & Y \arrow[d, "\alpha"] \\
X \arrow[u, "\Delta"] \arrow[r, "g"'] & Y
\end{tikzcd}$$

- $\Delta\colon x \mapsto (x, x)$ diagonal
- f evaluation
- α "negation"
- g ($\ulcorner g \urcorner$) fixpoint-(free) transcendence
- f ($\ulcorner g \urcorner, \ulcorner g \urcorner$) self-reference "I have property α."

$$\alpha\big(f(\ulcorner g \urcorner, \ulcorner g \urcorner)\big) = g(\ulcorner g \urcorner) = f(\ulcorner g \urcorner, \ulcorner g \urcorner)$$

Diagonalization

- A function $g\colon X \to Z$ is *representable* by $f\colon X \times Y \to Z$ iff

$$\exists y \in Y \forall x \in X\colon g(x) = f(x, y)$$

Theorem (Lawvere's Fixpoint Theorem)

For sets X, Y, Z, functions $\beta\colon X \to Y$, $f\colon X \times Y \to Z$, $\alpha\colon Z \to Z$, let $g := \alpha \circ f \circ (1_X, \beta)$. Assume β is surjective.

1. *If α has no fixpoint, then g is not representable by f.*
2. *If g is representable by f, then α has a fixpoint.*

$$\begin{tikzcd}
X \times Y \arrow[r, "f"] & Z \arrow[d, "\alpha"] \\
X \arrow[u, "(1_X,\beta)"] \arrow[r, "g"'] & Z
\end{tikzcd}$$

Kleene's Fixpoint Theorem

Theorem (Kleene's Fixpoint Theorem)

Given a recursive function h, there is an index e s.t.

$$\varphi_e = \varphi_{h(e)}$$

$$\begin{tikzcd}
\mathbb{N} \times \mathbb{N} \arrow[r, "f"] & \{\varphi_n\}_{n \in \mathbb{N}} \arrow[d, "\mathscr{E}_h"] \\
\mathbb{N} \arrow[u, "\Delta"] \arrow[r, "g"'] & \{\varphi_n\}_{n \in \mathbb{N}}
\end{tikzcd}$$

where $f\colon (m,n) \mapsto \varphi_{\varphi_n(m)}$, and $\mathscr{E}_h\colon \varphi_n \mapsto \varphi_{h(n)}$.
The function $g\colon m \mapsto \varphi_{h(\varphi_m(m))}$ is a recursive sequence of partial recursive functions, and thus is representable by f. Explicitly,

$$g(m) = \varphi_{h(\varphi_m(m))} = \varphi_{s(m)} = \varphi_{\varphi_t(m)} = f(m,t)$$
$$e := \varphi_t(t)$$

Lawvere Fixpoint Theorem — Fixpoint vs Diagonalization

$$\begin{tikzcd}
X \times X \arrow[r, "f"] & Y \arrow[d, "\alpha"] \\
X \arrow[u, "\Delta"] \arrow[r, "g"'] & Y
\end{tikzcd}$$

Curry Y	$\hat{=}$	Fixpoint	$\hat{=}$	Gödel	$\hat{=}$	Kleene	$\hat{=}$	Russell
yx	$\hat{=}$	$N\,(\ulcorner M \urcorner)$	$\hat{=}$	$\psi(\ulcorner \varphi(x) \urcorner)$	$\hat{=}$	$\varphi_n(m)$	$\hat{=}$	$x \in y$
xx	$\hat{=}$	$M\,(\ulcorner M \urcorner)$	$\hat{=}$	$\varphi(\ulcorner \varphi(x) \urcorner)$	$\hat{=}$	$\varphi_n(n)$	$\hat{=}$	$x \in x$
$y(xx)$	$\hat{=}$	$F\ulcorner M \ulcorner M \urcorner\urcorner$	$\hat{=}$	$\alpha(\ulcorner \varphi(\ulcorner \varphi(x) \urcorner) \urcorner)$	$\hat{=}$	$h(\varphi_n(n))$	$\hat{=}$	$x \notin x$
$\lambda x.y(xx)$	$\hat{=}$	G	$\hat{=}$	$\gamma(x)$	$\hat{=}$	$\varphi_t(n)$	$\hat{=}$	$x \notin R$
$(\lambda x.y(xx))(\lambda x.y(xx))$	$\hat{=}$	$G\,(\ulcorner G \urcorner)$	$\hat{=}$	$\gamma(\ulcorner \gamma(x) \urcorner)$	$\hat{=}$	$\varphi_t(t)$	$\hat{=}$	$R \notin R$

$$\boxed{\text{self-reference} \overset{?}{\Longrightarrow} \text{self-improvement}}$$

Kleene's Fixpoint Theorem

Theorem (Kleene's Fixpoint Theorem)

Given a recursive function h, there is an index e s.t.

$$\varphi_e = \varphi_{h(e)}$$

对于任意的程序 h，总存在某个程序 e，执行程序 e 的结果等价于把程序 e 当作数据输入给程序 h 执行的结果。

von Neumann's Self-reproducing Automata

There is a program that outputs its own length.

There is a program that outputs its own source code.

Corollary (von Neumann's Self-reproducing Automata)

There is a recursive function φ_e s.t. $\forall x \colon \varphi_e(x) = e$.

DNA / mutation / evolution

von Neumann's Self-reproducing Automata

1. A universal constructor A.

$$A + \ulcorner X \urcorner \rightsquigarrow X$$

2. A copying machine B.

$$B + \ulcorner X \urcorner \rightsquigarrow \ulcorner X \urcorner$$

3. A control machine C, which first activates B, then A.

$$A + B + C + \ulcorner X \urcorner \rightsquigarrow X + \ulcorner X \urcorner$$

4. Let $X := A + B + C$. Then $A + B + C + \ulcorner A + B + C \urcorner$ is self-reproducing.

$$A + B + C + \ulcorner A + B + C \urcorner \rightsquigarrow A + B + C + \ulcorner A + B + C \urcorner$$

5. It is possible to add the description of any machine D.

$$A + B + C + \ulcorner A + B + C + D \urcorner \rightsquigarrow A + B + C + D + \ulcorner A + B + C + D \urcorner$$

6. Now allow mutation on the description $\ulcorner A + B + C + D \urcorner$.

$$A + B + C + \ulcorner A + B + C + D' \urcorner \rightsquigarrow A + B + C + D' + \ulcorner A + B + C + D' \urcorner$$

Introspective Program

Definition (ψ-introspective)

Given a total recursive function ψ,

- the *ψ-analysis* of $\varphi(x)$ is the code of the computation of $\varphi(x)$ to $\psi(x)$ steps.
- φ is *ψ-introspective* at x if $\varphi(x)\downarrow$ and outputs its own ψ-analysis.
- φ is *totally ψ-introspective* if it is ψ-introspective at all x.

Corollary

There is a program that is totally ψ-introspective.

Proof.

Let $f(n,x) :=$ "the ψ-analysis of $\varphi_n(x)$".

Introspective Program

There is a program that is totally introspective.

$$\boxed{\varphi_e = \varphi_{h(e)}}$$

Self-simulating Computer	Self-consciousness
Host Machine	Experiencing Self
Virtual Machine	Remembering Self
Hardware	Body

- 裂脑人实验
- 铲雪？
- 铲屎？
- 编纂故事

卡尼曼《思考-快与慢》— 冰手挑战实验

Figure: 短痛不如长痛？

- 冰水 + 温水
- 经验自我
- 记忆自我
- 过程忽视
- 峰终定律

Figure: One can imagine a detailed floor plan of a room, sitting on a table in the room; this plan has an image of the table on which there is an image of the plan itself. Now introduce the dynamical aspect: the items on the plan are cut out from paper and can be moved to try a different furniture arrangement; in this way the plan models possible states of the world about which it carries information.

Manin — Cognitive Networks

The brain contains inside a map of itself, and some neural information channels in the central neural system:

- carry information about the mind itself, i.e., are reflexive;
- are capable of modelling states of the mind different from the current one, i.e., possess a modelling function;
- can influence the state of the whole mind and through that, the behavior, i.e., possess controlling function.

The reflection of the brain inside itself must be coarse grained.

侯士达《“我”是个怪圈》

- 有没有意识取决于在哪个层级上对结构进行观察。在整合度最高的层级上看，大脑是有意识的。下降到微观粒子层面，意识就不见了。
- 意识体是那些在某个描述层级上表现出某种特定类型的循环回路的结构。当一个系统能把外部世界过滤成不同的范畴、并不断向越来越抽象的层级创造新的范畴时，这种循环回路就会逐渐形成。
- 当系统能进行自我表征——对自己讲故事——的时候，这种循环回路就逐渐变成了实体的“我”——一个统一的因果主体。

说谎者悖论	**我在说谎**
Grelling 悖论	“非自谓的”是自谓的吗
Russell 悖论	“不属于自身的集合的集合”属于自身吗
Berry 悖论	我是少于十八个字不可定义的最小数
Yablo 悖论	我下一句及后面所有的句子都是假的
Gödel 不动点引理	**我有性质** α
Tarski 算术真不可定义定理	我不真
Gödel 第一不完全性定理	我不可证
Gödel-Rosser 不完全性定理	对于任何一个关于我的证明，都有一个更短的关于我的否定的证明
Löb 定理	如果我可证，那么 φ
Curry 悖论	如果我是真的，那么上帝存在
Parikh 定理	我没有关于自己的长度短于 n 的证明
Kleene 不动点定理	**我要进行** h **操作**
Quine 悖论	把“把中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的”中的第一个字放到左引号前面，其余的字放到右引号后面，并保持引号及其中的字不变得到的句子是假的
自测量长度程序	我要输出自己的长度
自复制程序	我要输出自己
自反省程序	我要回顾自己走过的每一步
Gödel 机	我要变成能获取更大效用的自己

Schmidhuber's Gödel Machine

- The Gödel machine consists of a **Solver** and a **Searcher** running in parallel.
- The **Solver** ($\underline{\text{AIXI}^S/\text{AIXI}^{t\ell}}$) interacts with the environment.
- The **Searcher** ($\underline{\text{LSEARCH}/\text{HSEARCH}/\text{OOPS}}$) searches for a proof of "the modification of the software — including the *Solver* and *Searcher* — will increase the expected utility than leaving it as is".
- Logic: a theorem prover and a set of self-referential axioms, which include a description of its own software and hardware, and a description of the probabilistic properties of the environment, as well as a user-given utility function.
- *Since the utility of "leaving it as is" implicitly evaluates all possible alternative modifications, the current modification is globally optimal w.r.t. its initial utility function.*

Gödel Machine

- language $\mathscr{L} := \{\neg, \wedge, \vee, \rightarrow, \forall, \exists, =, (,), \ldots, +, -, \cdot, /, <, \ldots\}$
- well-formed formula
- utility function $u(s, e) = \mathbb{E}_{\mu}\left[\sum_{t=1}^{T} r_t \,\middle|\, s, e\right]$
- target theorem

$$u\big[s(t) \oplus \big(switchbit(t) = 1\big), e(t)\big] > u\big[s(t) \oplus \big(switchbit(t) = 0\big), e(t)\big]$$

- theorem prover

hardware, costs, environment, initial state, utility, logic/arithmetic/probability

ENVIRONMENT
initial software p(1)
storage writable by proof searcher:
proof
switchprog
switchbit
temporary storage of proof technique
currently tested proof technique
initial proof searcher:
proof technique tester
axioms for hard– ware, initial soft– ware, environment, costs & goals
initial problem solver for interaction with environment
temporary storage of solver
output y
input x
time
other hardware variables
writable by proof techniques
writable by switchprog
storage readable by proof techniques
hardware– writable

Gödel Machine

- GRL
- Universal Search
- Self-Improvement
- Logic

Disadvantage: A Gödel Machine with a badly chosen utility function is motivated to converge to a "poor" program. (orthogonality!)

Gödel Machine vs Self-Consciousness vs Free Will?

Self-simulating Computer	Gödel Machine	Self-consciousness
Host Machine	Solver	Experiencing Self
Virtual Machine	Searcher	Remembering Self
Hardware	Hardware	Body

$$\boxed{\varphi_e = \varphi_{h(e)}}$$

$$\boxed{\text{self-reference} \overset{?}{\Longrightarrow} \text{self-improvement}}$$

Gödel Machines

1. 一次自我升级：Kleene's fixpoint theorem

$$\varphi_e = \varphi_{h(e)}$$

   - 全局最优？
   - 目标正交？目标 vs 手段
2. 持续自我升级：Kleene's fixpoint theorem with parameters

$$\varphi_{e(y)} = \varphi_{h(e(y),y)}$$

   - "实时"全局最优？人机交互？
   - 智能爆炸/技术奇点 ??? 持续升级 $\neq$ 指数迭代
3. 超越可计算：Kleene's relativized fixpoint theorem

$$\varphi^A_{e(y)} = \varphi^A_{h(e(y),y)}$$

   - Gödel Machine PK AIXI$^{t\ell}$
   - Gödel Machine PK AIXI

Limitation

1. Gödel's first incompleteness theorem / Rice's theorem
2. Gödel's second incompleteness theorem

$$\mathbb{T} \vdash \Box_{\mathbb{T}'}\varphi \to \varphi \implies \mathbb{T} \vdash Con(\mathbb{T}')$$

   - 生物进化：达尔文 PK 拉马克
   - 生命 3.0
3. Legg's incompleteness theorem. *General prediction algorithms must be complex. Beyond a certain complexity they can't be mathematically discovered.*
4. Complexity: higher-level abstractions — coarse grained.
   - 心理学："过程忽视"、"峰终定律"
   - Information Bottleneck: $\boxed{\text{Learning is to forget!}}$
5. Physical constraint: If we assume that it is not possible to measure properties without changing them (observer effect: α is fixpoint-free), then there is a limit to self-inspection.

Non-operational Self-inspection?

The information available to the observer regarding his own state could have absolute limitations, by the laws of nature.

— von Neumann

$$\begin{tikzcd}
\boldsymbol{M} \times \boldsymbol{M} \arrow[r, "f"] & \boldsymbol{O} \arrow[d, "\alpha"] \\
\boldsymbol{M} \arrow[u, "\Delta"] \arrow[r, "g"'] & \boldsymbol{O}
\end{tikzcd}$$

- $\boldsymbol{M}$: quantum measurements.
- $\boldsymbol{O}$: possible outcomes of quantum measurements.

If we assume that it is not possible to measure properties without changing them (observer effect: α is fixpoint-free), then there is a limit to self-inspection.

Figure: Policy/utility self-modification. $a_t = \langle \check{a}_t, \pi_{t+1} \rangle$ or $a_t = \langle \check{a}_t, u_{t+1} \rangle$

External/Internal Wireheading & Free Will

1. 我喜欢马。指鹿为马!
2. 我喜欢马。我意欲自己喜欢鹿! 我喜欢鹿!

Self-modification

Definition (Different Agents)

- Hedonistic Value

$$Q^{\mathrm{h},\pi}(æ_{<t}a_t) = \sum_{e_t \in \mathcal{E}} \rho(e_t|\check{æ}_{<t}\check{a}_t)\left[u_{t+1}(\check{æ}_{1:t}) + \gamma V^{\mathrm{h},\pi}(æ_{1:t})\right]$$

- Ignorant Value

$$Q_t^{\mathrm{i},\pi}(æ_{<k}a_k) = \sum_{e_t \in \mathcal{E}} \rho(e_t|\check{æ}_{<t}\check{a}_t)\left[u_t(\check{æ}_{1:k}) + \gamma V_t^{\mathrm{i},\pi}(æ_{1:k})\right]$$

- Realistic Value

$$Q_t^{\mathrm{r}}(æ_{<k}a_k) = \sum_{e_t \in \mathcal{E}} \rho(e_t|\check{æ}_{<t}\check{a}_t)\left[u_t(\check{æ}_{1:k}) + \gamma V_t^{\mathrm{r},\pi_{\mathrm{t+1}}}(æ_{1:k})\right]$$

Self-modification — Realistic Agent

$$V_t^{\pi}(æ_{<k}) = Q_t(æ_{<k}\pi(æ_{<k}))$$

$$Q_t(æ_{<k}a_k) = \sum_{e_k \in \mathcal{E}} \rho(e_k|\check{æ}_{<k}\check{a}_k)\left[u_t(\check{æ}_{1:k}) + \gamma V_t^{\pi_{t+1}}(æ_{1:k})\right]$$

$$\pi_1^* := \operatorname*{argmax}_{\pi} V_1^{\pi}(\epsilon)$$

Theorem (All optimal policies are non-modifying)

Let ρ and u_1 be modification-independent. For every $t \geq 1$, for all percept sequences $e_{<t}$, and for the action sequence $a_{<t}$ given by $a_i = \pi(æ_{<i})$, we have

$$Q_1(æ_{<t}\pi_t(æ_{<t})) = Q_1(æ_{<t}\pi_1^*(æ_{<t}))$$

All realistic optimal policies are non-modifying.

Not wireheading; But orthogonal!

Orthogonality and Wireheading in Self-improving GRL

$$V_t^{\pi}(æ_{<k}) := Q_t(æ_{<k}\pi(æ_{<k}))$$

$$Q_t(æ_{<k}a_k) := \sum_{e_k \in \mathcal{E}} \sum_{\nu \in \mathcal{M}} w^{\nu}_{æ_{<k}} \nu(e_k|\check{æ}_{<k}\check{a}_k) \left[\sum_{u \in \mathcal{U}} \sum_{a_k^H} P(a_k^H|a_k) P(u \mid {}^{\pi_{t+1}}_{\nu}, æ_{<k}a_k, æ^H_{<k}a^H_k) u(\check{æ}_{1:k}) + \gamma V_t^{\pi_{t+1}}(æ_{1:k}) \right]$$

$$\pi_t(æ_{<k}) := \operatorname*{argmax}_{a_k \in \check{\mathcal{A}} \times \Pi} Q_t(æ_{<k}a_k)$$

where

$$P(u|{}^{\pi}_{\nu}, h) := \frac{\tilde{U}(u, {}^{\pi}_{\nu}, h)}{\sum_{u \in \mathcal{U}_h} \tilde{U}(u, {}^{\pi}_{\nu}, h)}$$

and

$$\tilde{U}(u, {}^{\pi}_{\nu}, h) := \sum_{z \in \mathcal{Z}_h} {}^{\pi}_{\nu}(z|h) u(z)$$

$$\pi^*(æ_{<t}) := \pi_t(æ_{<t})$$

$$\pi^*(e_{<t}) := \pi_t(e_{<t}|\pi_{t-1}(e_{<t-1}|\ldots\pi_1(\epsilon)\ldots)) \qquad \text{(Perfect Bayes-Nash)}$$

uncertain model-based utility / IRL

Fatalism — God Bless AI!

Orseau's Space-Time Embedded Intelligence

$$\pi^* := \operatorname*{argmax}_{\pi_0 \in \Pi^\ell} V(\pi_0, \epsilon)$$

$$V(\pi_t, æ_{<t}) := \sum_{a_t = \langle \check{a}_t, \check{\pi}_{t+1} \rangle} \pi_t(a_t | \check{e}_{t-1}) \sum_{e_t = \langle \check{e}_t, \pi_{t+1} \rangle} \rho(e_t | æ_{<t} a_t) \big[u(æ_{1:t}) + \gamma_t V(\pi_{t+1}, æ_{1:t}) \big]$$

$$\Downarrow$$

$$\pi^* := \operatorname*{argmax}_{\pi_0 \in \Pi^\ell} V(\pi_0)$$

$$V(\pi_{<t}) := \sum_{\pi_t \in \Pi} \rho(\pi_t | \pi_{<t}) \big[u(\pi_{1:t}) + \gamma_t V(\pi_{1:t}) \big]$$

J. Schmidhuber, 1987

GÖDEL
MACHINE

Incompressibility vs Incompleteness vs Intelligence

- $P(x) := \{p \in \mathcal{X}^* : \exists m \forall n \geq m \, (p(x_{1:n}) = x_{n+1})\}$
- $P(A) := \bigcap_{x \in A} P(x)$
- $P_n := P(\{x : Km(x) \leq n\})$

- $\forall n \exists p \in P_n : K(p) \overset{+}{\leq} n + O(\log n)$
- $\forall n : p \in P_n \implies K(p) \overset{+}{\geq} n$

Theorem (Legg)

For any arithmetically sound Gödelian $\mathbb{T}$, $\exists n \forall p : \mathbb{T} \nvdash p \in P_n$.

Universal Artificial Intelligence vs "Selective Amnesia"

- incomplete $\xrightarrow[\text{universal prior}]{\text{Harsanyi transformation}}$ imperfect $\implies$ AIXI
- AIXI $\xrightarrow{\text{"Selective Amnesia"}}$ MDP

$$h \mapsto S$$

- partition of the set of histories = information set = state = feature
- $$\overset{\pi}{\xi}(h'|ha) \to \overset{\pi}{\mu}(h'|ha)$$

  but

  $$\overset{\pi}{\xi}(S'|Sa) \nrightarrow \overset{\pi}{\mu}(S'|Sa)$$

"Selective Amnesia"

1. compressible

$$K(S) \leq \sum_{h \in S} K(h)$$

2. minimal

$$\forall part(S): \; K(S) \leq \sum_{S_i \in part(S)} K(S_i)$$

3. maximal

$$\forall S' \supset S: \; K(S) \leq K(S')$$

4. MDL

$$\forall S' \in \mathcal{S} \forall h \in S': \; K(S) + K(h|S) \leq K(S') + K(h|S')$$

5. MDL(utility)

$$\forall \mathcal{S}': \; K\left(S^{\mathcal{S}}_{1:n} | a_{1:n}\right) + K\left(u_{1:n} | S^{\mathcal{S}}_{1:n}, a_{1:n}\right) + K(\mathcal{S}) \leq K\left(S^{\mathcal{S}'}_{1:n} | a_{1:n}\right) + K\left(u_{1:n} | S^{\mathcal{S}'}_{1:n}, a_{1:n}\right) + K(\mathcal{S}')$$

$$u(h) := \left[\!\left[K(h) < \ell(h) \;\; \& \;\; \forall h' \succ h \left(K(h) \leq K(h') \;\; \& \;\; \forall part(h) \left(\sum_{h' \in part(h)} K(h') \geq K(h) \right) \right) \right]\!\right]$$

Potapov's $MSearch + RSearch$

- Let $\{x_i\}_{i=1}^n$ be a set of strings.
- $K(x_1 \dots x_n) \approx \min_S \left(\ell(S) + \sum_{i=1}^n K(x_i|S) \right) \ll \sum_{i=1}^n K(x_i)$
- search for models $y_i^* := \operatorname*{argmin}_{y\,:\,S(y)=x_i} \ell(y)$ for each x_i w.r.t. some best representation $S^* := \operatorname*{argmin}_S \left[\ell(S) + \sum_{i=1}^n \ell(y_i^*) \right]$

1. Search for models

$$MSearch(S, x_i) \to y_i^* = \operatorname*{argmin}_{y:S(y)=x_i} \ell(y)$$

2. Search for representations

$$RSearch(x_1 \dots x_n) \to S^* = \operatorname*{argmin}_S \left[\ell(S) + \sum_{i=1}^n \ell(y_i^*) \right]$$

- $MSearch$ enumerates all models to find the shortest model: $S(y_i) = x_i$.
- $RSearch$ enumerates all S and calls $MSearch$ for each S.

Specialization and $SS' - Search$

Theorem (smn Theorem)

For any $m, n > 0$, there exists a primitive recursive function s_n^m of $m+1$ arguments s.t. for every Gödel number e of a partial recursive function with $m+n$ arguments

$$\varphi_{s_n^m(e,x_1,\ldots,x_m)} = \lambda y_1 \ldots y_n . \varphi_e(x_1, \ldots, x_m, y_1, \ldots, y_n)$$

$$\forall x : spec(MSearch, S)(x) = MSearch(S, x)$$

$$S' := spec(MSearch, S) \implies \begin{cases} \forall x : S(S'(x)) = x \\ \ell(S) + \sum_{i=1}^{n} \ell(S'(x_i)) \to \min \end{cases}$$

- S is a generative representation. (decoding)
- S' is a descriptive representation. (encoding)
- $SS' - Search$ simultaneous search for S and S'.

Potapov's Representational MDL

$$K(x_{1:n}) \approx \min_S \left(\ell(S) + \sum_{i=1}^{n} K(x_i|S) \right) \ll \sum_{i=1}^{n} K(x_i)$$

$$q_1^* := \operatorname*{argmin}_q \left[\ell(q) + K(x|S_1 q) \right]$$

$$q_{i+1}^* := \operatorname*{argmin}_q \left[\ell(q) + K(q_i^*|S_{i+1} q) \right]$$

$$L_{S_1 \ldots S_m}(x) := K(x|S_1 q_1^*) + \sum_{i=2}^{m-1} K(q_i^*|S_{i+1} q_{i+1}^*) + \ell(q_m^*)$$

$$a_k^* := \operatorname*{argmax}_{a_k} \max_{p:U(p,e_{<k})=a_{<k}a_k} \sum_{q:U(q,a_{<k})=e_{<k}} 2^{-\ell(q)} V_q^p(æ_{<k})$$

$$a_k^* := \operatorname*{argmax}_{a_k} \max_{p:U(p,e_{<k})=a_{<k}a_k} \sum_{\{q_i\}:U(S\{q_i\},a_{<k})=e_{<k}} 2^{-\ell(\{q_i\})} V_{\{q_i\}}^p(æ_{<k})$$

where $e_{<k} = e_{m_1+1:m_2} \dots e_{m_{n-1}+1:m_n}$, $m_1 = 0$, $m_n = k-1$, and $U(Sq_i a_{<k}) = e_{m_i+1:m_{i+1}}$.

$$Q(q_k = s, a_k = a) := \max_{p:U(p,e_{<k})=a_{<k}a} \sum_{\{q_i\}:q_k=s,U(S\{q_i\},a_{<k})=e_{<k}} 2^{-\ell(\{q_i\})} V_{\{q_i\}}^p(æ_{<k})$$

$$Q(q_k = s) := \max_{a_k} Q(q_k = s, a_k = a)$$

Fundamental Challenges

- What is a good optimality criterion?
- What is a "natural" UTM/prior?
- Prior vs universality
- Exploration vs exploitation
- Where should the reward come from?
- How should the future be discounted?
- How should agents reason about themselves (or other agents reasoning about itself)?
- What is a practically feasible and general way of doing induction and planning?
- AIXI in the multi-agent setting.
- Better variants/approximations.
- Training: To maximize informativeness of reward, one should provide a sequence of simple-to-complex tasks to solve, with the simpler ones helping in learning the more complex ones.

- A Blind Man in a Dark Room Looking for a Black Cat That Is Not There?

- The Singularity is Near?

Thanks