Universal Artificial Intelligence

Sequential Decisions Based on Algorithmic Probability

adapted by Łukasz Stafiniak

from the book by Marcus Hutter

Table of contents

Universal Sequence Prediction 5

Ockham, Epicurus, Hume, Bayes, Solomonoff 5
Algorithmic Information Theory and Probability 6
Convergence, Error and Loss Bounds 11
Convergence 13
Error bounds 14
Loss bounds 15
Games of Chance 18
Optimality 19

The Universal Algorithmic Agent AIXI 21

Agents in Known Probabilistic Environments 21
AIμ model in Functional Form 22
AIμ model in Recursive and Iterative Form 24
Factorizable environments μ. 26
Probabilistic policies. 27
Persistent Turing Machines ⟨reference|PTMs⟩ 28
The Universal Algorithmic Agent AIXI 29

Intelligence Order Relation. 30
Separability concepts. 31
Value-Related Optimality, Discounted Future Value Function 33
The Choice of the Horizon. 38
Actions as random variables 40
Uniform mixture of MDPs 41
Important Environmental Classes 43
Sequence Prediction 43
Strategic Games 44
Using the AIξ model for game playing. 45
Function Minimization 46
Supervised Learning from Examples (EX) 48

AIXItl and Optimal Search 49

The Fastest and Shortest Algorithm for All Problems 49
Levin Search 50
The Fast Algorithm $M_{p^*}^{\varepsilon}$ 51
Time-Bounded AIXI Model 52
Time-Limited Probability Distributions. 52
Best Vote Algorithm. 52

The Universal Time-Bounded AIXItl Agent. 53
Optimality of AIXItl. 54
Speed Prior and OOPS ⟨reference|TheNewAI⟩ 56
Speed Prior 56
Optimal Ordered Problem Solver 57
Goedel Machine ⟨reference|GoedelMachines⟩ 60
A program that prints itself. 61
Proof Techniques 62
Possible Types of Goedel Machine Self-improvements 63

Universal Sequence Prediction

Ockham, Epicurus, Hume, Bayes, Solomonoff

- Epicurus' principle of multiple explanations
- Occam's razor (simplicity) principle
- Hume's negation of induction
- Bayes' rule for conditional probabilities
- Solomonoff's universal theory of inductive inference

Induction here = reasoning about future from past experience.
Prequential approach (transductive inference) = predictions without building a model.

- Every induction problem can be phrased as sequence prediction task.
  - Classification is a special case of sequence prediction.
- We are interested in maximizing profit / minimizing loss.
- Separating noise from data is not necessary.

Algorithmic Information Theory and Probability

- A prefix code (prefix-free set of strings) satisfies:

$$\sum_{x \in \mathcal{P}} 2^{-l(x)} \leqslant 1$$

- Kolmogorov complexity:

$$K(x) := \min_p \{l(p) : U(p) = x\}, K(x|y) := \min_p \{l(p) : U(y'p) = x\}$$

Properties:

  - $K(x) \stackrel{+}{\leqslant} l(x) + 2\log_2 l(x)$
  - $K(x, y) \stackrel{+}{=} K(y|x, K(x)) + K(x)$
  - $K(x) \stackrel{+}{\leqslant} -\log_2 P(x) + K(P)$ if $P : \mathbb{B}^* \to [0, 1]$ enum. and $\sum_x P(x) \leqslant 1$

Kolmogorov complexity is only co-enumerable=upper semi-computable.

- Bayes rule:

$$p(H_i|D) = \frac{p(D|H_i)p(H_i)}{\sum_{i \in I} p(D|H_i)p(H_i)}$$

- Universal prior: probability that the output of a universal monotone TM starts with x when provided with fair coin flips on the input tape

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

- $\mu \geqslant 0$ is a semimeasure if $\mu(\epsilon) \leqslant 1$ and $\mu(x) \geqslant \mu(x0) + \mu(x1)$ (a probability measure if equality holds).

- **Universality of M.** M multiplicatively dominates all enumerable semimeasures:

$$M(x) \overset{+}{\geqslant} 2^{-K(\rho)}\, \rho(x)$$

where ρ is an enumerable semimeasure. M is enumerable but not estimable. Conditioning on a string:

$$M(y|x) := \frac{M(x\,y)}{M(x)} \approx 2^{-K(y|x)}$$

- Try to predict the continuation $x_n \in \mathbb{B}$ of a given sequence $x_1 \ldots x_{n-1}$.

$$\sum_{t=1}^{\infty} (1 - M(x_t|x_{<t}))^2 \leqslant -\frac{1}{2}\sum_{t=1}^{\infty} \ln M(x_t|x_{<t}) = -\frac{1}{2}\ln M(x_{1:\infty})$$
$$\leqslant \frac{1}{2}\ln 2\, \mathrm{Km}(x_{1:\infty})$$

If $x_{1:\infty}$ is computable, then $\mathrm{Km}(x_{1:\infty}) < \infty$, and $M(x_t|x_{<t}) \to 1$.

- Assume now the true sequence is drawn from a computable probability distribution μ. The probability of x_n given $x_{<n}$ is $\mu(x_n|x_{<n}) = \mu(x_{1:n})/\mu(x_{<n})$.

Posterior convergence of M to μ:

$$\sum_{t=1}^{\infty} \sum_{x_{<t} \in \mathbb{B}^{t-1}} \mu(x_{<t})(M(0|x_{<t}) - \mu(0|x_{<t}))^2 \stackrel{+}{\leqslant} \frac{1}{2}\ln 2\, K(\mu) < \infty$$

That is, $M(0|x_{<t}) - \mu(0|x_{<t})$ tends to zero with μ-probability 1. We will see a proof later, by approximating M with

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

- A sequence is μ-Martin-Loef random (μ.M.L.) iff:

$$\exists c \forall n \, M(x_{1:n}) \leqslant c \, \mu(x_{1:n})$$

is μ/ξ-Martin-Loef random ($\mu.\xi$.r.) iff:

$$\exists c \forall n \, \xi(x_{1:n}) \leqslant c \, \mu(x_{1:n})$$

  - A theorem true for all μ-M.L. random sequences is true with μ-probability 1.

- **Complexity increase**:

  - $K(y\,x) - K(y) \leqslant K(x|y) + O(1)$ (prefix Kolmogorov complexity)
  - $C(y\ x) - C(y) \leqslant \tilde{K}(x|y) + O(K(C(y))$ (plain Kolmogorov complexity)
  - $\mathrm{KM}(y\,x) - \mathrm{KM}(y) \leqslant \tilde{K}(x|y) + O(K(l(y))$, $\mathrm{KM}(x) := -\log_2 M(x)$
  - $\mathrm{KM}(y\,x) - \mathrm{KM}(y) \leqslant \tilde{K}(\mu|y) - \log_2 \mu(x|y) + O(K(l(y))$

where $\tilde{K} \in \{K, C\}$.

- Predictor based on K fails, due to $K(x1) \stackrel{+}{=} K(x0)$. The monotone complexity $\mathrm{Km}(x) := \min_p \{l(p)\colon U(p) = x\,*\}$ does not suffer from this. $m(x) := 2^{-\mathrm{Km}(x)}$ is extremely close to $M(x)$. $m = 2^{-\mathrm{Km}}$
  - converges rapidly on-sequence

$$\sum_{t=1}^{n} \left| 1 - m(x_t|x_{<t}) \right| \leqslant \frac{1}{2}\mathrm{Km}(x_{1:n})$$

    $m(x_t|x_{<t}) \neq 1$ at most $\mathrm{Km}(x_{1:\infty})$ times
  - may converge slowly off-sequence

$$\sum_{t=1}^{n} m(\bar{x}_t|x_{<t}) \leqslant 2^{\mathrm{Km}(x_{1:n})}$$

$$\forall s \exists U, x_{1:\infty}\colon \mathrm{Km}(x_{1:\infty}) = s \wedge \sum_{t=1}^{\infty} m(\bar{x}_t|x_{<t}) \geqslant 2^s - 2$$

  - may not converge for probabilistic environments

$$\exists \mu \in \mathcal{M}^{\mathrm{msr}}_{\mathrm{comp}} \setminus \mathcal{M}_{\mathrm{det}}\colon m(x_t|x_{<t}) \stackrel{t\to\infty}{\not\longrightarrow} \mu(x_t|x_{<t}) \forall x_{1:\infty}$$

m is not a semimeasure, but normalization does not improve the above.

Convergence, Error and Loss Bounds

- Assumptions: a mixture distribution ξ is a w_ν-weighted sum of probability distributions ν of a set $\mathcal{M}$ containing the true distribution μ.

$$\xi(x_t|x_{<t}) = \sum_{\nu\in\mathcal{M}} w_\nu(x_{<t})\nu(x_t|x_{<t}), w_\nu(x_{1:t}) := w_\nu(x_{<t})\frac{\nu(x_t|x_{<t})}{\xi(x_t|x_{<t})}$$

  and $w_\nu(\epsilon) = w_\nu$.

- Distance measures:
  - absolute (or Manhattan): $a(\boldsymbol{y},\boldsymbol{z}) := \sum_i \left|y_i - z_i\right|$
  - quadratic (or squared Euclidean): $s(\boldsymbol{y},\boldsymbol{z}) := \sum_i \left(y_i - z_i\right)^2$
  - (squared) Hellinger distance: $h(\boldsymbol{y},\boldsymbol{z}) := \sum_i \left(\sqrt{y_i} - \sqrt{z_i}\right)^2$
  - relative entropy or KL divergence: $d(\boldsymbol{y},\boldsymbol{z}) := \sum_i y_i \ln\frac{y_i}{z_i}$
  - absolute divergence: $b(\boldsymbol{y},\boldsymbol{z}) := \sum_i y_i \left|\ln\frac{y_i}{z_i}\right|$
- Entropy inequalities:
  - $\frac{1}{2}\sum f \leqslant f\sqrt{\frac{1}{2}d}$
  - $s \leqslant d$
  - $b - d \leqslant a \leqslant \sqrt{2d}$
  - $h \leqslant d$

- Instantaneous (at time t) and total distances between μ and ξ:

$$\mathcal{X}=\{1,...,N\}, N=|\mathcal{X}|, i=x_t, y_i=\mu(x_t|x_{<t}), z_i=\xi(x_t|x_{<t})$$

  - $a_t(x_{<t}) := \sum_{x_t} \left|\mu(x_t|x_{<t}) - \xi(x_t|x_{<t})\right|, A_n := \sum_{t=1}^n \boldsymbol{E}[a_t(x_{<t})]$
  - $s_t(x_{<t}) := \sum_{x_t} (\mu(x_t|x_{<t}) - \xi(x_t|x_{<t}))^2, S_n := \sum_{t=1}^n \boldsymbol{E}[s_t(x_{<t})]$
  - $h_t(x_{<t}) := \sum_{x_t} \left(\sqrt{\mu(x_t|x_{<t})} - \sqrt{\xi(x_t|x_{<t})}\right)^2, H_n := \sum_{t=1}^n \boldsymbol{E}[h_t(x_{<t})]$
  - $d_t(x_{<t}) := \sum_{x_t} \mu(x_t|x_{<t}) \ln \frac{\mu(x_t|x_{<t})}{\xi(x_t|x_{<t})}, D_n := \sum_{t=1}^n \boldsymbol{E}[d_t(x_{<t})]$
  - $b_t(x_{<t}) := \sum_{x_t} \mu(x_t|x_{<t}) \left|\ln \frac{\mu(x_t|x_{<t})}{\xi(x_t|x_{<t})}\right|, B_n := \sum_{t=1}^n \boldsymbol{E}[b_t(x_{<t})]$

- For example the first convergence result on next page says:

$$\sum_{t=1}^n \boldsymbol{E}\left[\sum_{x_t} (\mu(x_t|x_{<t}) - \xi(x_t|x_{<t}))^2\right] \leqslant \ln w_\mu^{-1}$$

- **Convergence** **Convergence of ξ to μ:**

  - $S_n \leqslant D_n \leqslant \ln w_\mu^{-1} < \infty$
  - $s_t(x_{<t}) \leqslant d_t(x_{<t}) \stackrel{t\to\infty}{\longrightarrow} 0$ w.μ.p.1
  - $\xi(x_t'|x_{<t}) - \mu(x_t'|x_{<t}) \stackrel{t\to\infty}{\longrightarrow} 0$ w.μ.p.1 (and i.m.s.) for any x_t'
  - $\sum_{t=1}^n \boldsymbol{E}\left[\left(\sqrt{\frac{\xi(x_t|x_{<t})}{\mu(x_t|x_{<t})}} - 1\right)^2\right] \leqslant H_n \leqslant D_n \leqslant \ln w_\mu^{-1} < \infty$
  - $\sqrt{\frac{\xi(x_t|x_{<t})}{\mu(x_t|x_{<t})}} \stackrel{t\to\infty}{\longrightarrow} 1$ i.m.s. and $\frac{\xi(x_t|x_{<t})}{\mu(x_t|x_{<t})} \stackrel{t\to\infty}{\longrightarrow} 1$ w.μ.p.1
  - $b_t(x_{<t}) - d_t(x_{<t}) \leqslant a_t(x_{<t}) \leqslant \sqrt{2d_t(x_{<t})}$, $B_n - D_n \leqslant A_n \leqslant \sqrt{2n\,D_n}$

  where w_μ is the weight of μ in ξ and $x_{1:\infty}'$ is arbitrary (nonrandom) sequence.

- μ/ξ-randomness cannot be decided from ξ being a mixture distribution and the dominance property alone. (E.g., for Bernoulli sequences, it is related to denseness of $\mathcal{M}_\Theta$.)

- When $\mu \notin \mathcal{M}$, but $\hat{\mu} \in \mathcal{M}$ with the KL divergence $D_n(\mu\|\hat{\mu}) := \sum_{x_{1:n}} \mu(x_{1:n}) \ln \frac{\mu(x_{1:n})}{\hat{\mu}(x_{1:n})} \leqslant c$, then

$$D_n = \boldsymbol{E}\left[\ln \frac{\mu(x_{1:n})}{\xi(x_{1:n})}\right] = \boldsymbol{E}\left[\ln \frac{\hat{\mu}(x_{1:n})}{\xi(x_{1:n})}\right] + \boldsymbol{E}\left[\ln \frac{\mu(x_{1:n})}{\hat{\mu}(x_{1:n})}\right] \leqslant \ln w_{\hat{\mu}}^{-1} + c$$

- **Error bounds**

  A prediction scheme Θ_ρ predicts $x_t^{\Theta_\rho} := \mathrm{argmax}_{x_t} \rho(x_t|x_{<t})$. Probability of making wrong prediction and the expected number of errors:

$$e_t^{\Theta_\rho}(x_{<t}) := 1 - \mu(x_t^{\Theta_\rho}|x_{<t}), E_n^{\Theta_\rho} := \sum_{t=1}^{n} \boldsymbol{E}\left[e_t^{\Theta_\rho}(x_{<t})\right]$$

- **Error bound**:

$$\begin{aligned} 0 \leqslant E_n^{\Theta_\xi} - E_n^{\Theta_\mu} &\leqslant \sqrt{2\left(E_n^{\Theta_\xi} + E_n^{\Theta_\mu}\right) S_n} \\ &\leqslant S_n + \sqrt{4 E_n^{\Theta_\mu} S_n + S_n^2} \\ &\leqslant 2S_n + 2\sqrt{E_n^{\Theta_\mu} S_n} \end{aligned}$$

- **Loss bounds**

  Let $l_{x_t y_t} \in \mathbb{R}$ be the received loss when taking action $y_t \in \mathcal{Y}$ and $x_t \in \mathcal{X}$ is the tth symbol of the sequence. W.l.o.g., $0 \leqslant l_{x_t y_t} \leqslant 1$. We name an action a prediction even if $\mathcal{X} \neq \mathcal{Y}$.

- A prediction scheme Λ_ρ predicts

$$y_t^{\Lambda_\rho} := \arg\min_{y_t \in \mathcal{Y}} \sum_{x_t} \rho(x_t | x_{<t}) l_{x_t y_t}$$

  The actual and total μ-exptected loss:

$$l_t^{\Lambda_\rho}(x_{<t}) := \boldsymbol{E}_t\Big[l_{x_t y_t^{\Lambda_\rho}}\Big], L_n^{\Lambda_\rho} := \sum_{t=1}^{n} \boldsymbol{E}\Big[l_t^{\Lambda_\rho}(x_{<t})\Big]$$

- **Unit loss bound**:

$$\begin{aligned} 0 \leqslant L_n^{\Lambda_\xi} - L_n^{\Lambda_\mu} &\leqslant D_n + \sqrt{4L_n^{\Lambda_\mu} D_n + D_n^2} \\ &\leqslant 2D_n + 2\sqrt{E_n^{\Lambda_\mu} D_n} \end{aligned}$$

Collorary:

- $L_\infty^{\Lambda_\xi}$ is finite $\iff$ $L_\infty^{\Lambda_\mu}$ is finite,
- $L_\infty^{\Lambda_\xi} \leqslant 2D_\infty \leqslant 2\ln w_\mu^{-1}$ for deterministic μ if $\forall x \exists y\, l_{xy} = 0$,
- $L_n^{\Lambda_\xi} / L_n^{\Lambda_\mu} = 1 + O\Big((L_n^{\Lambda_\mu})^{-1/2} \Big) \overset{L_n^{\Lambda_\mu} \to \infty}{\longrightarrow} 1$
- $L_n^{\Lambda_\xi} - L_n^{\Lambda_\mu} = O\Big(\sqrt{L_n^{\Lambda_\mu}} \Big)$

Let Λ be any prediction scheme:

- $L_n^{\Lambda_\mu} \leqslant L_n^{\Lambda}$, $l_t^{\Lambda_\mu}(x_{<t}) \leqslant l_t^{\Lambda}(x_{<t})$
- $L_n^{\Lambda} \geqslant L_n^{\Lambda_\xi} - 2\sqrt{L_n^{\Lambda_\xi} D_n}$
- $L_n^{\Lambda_\xi} / L_n^{\Lambda} \leqslant 1 + O\Big((L_n^{\Lambda})^{-1/2} \Big)$

- $l_t^{\Lambda_\xi} - l_t^{\Lambda_\mu} \to 0$ would follow from $\xi \to \mu$ by continuity, but $l_t^{\Lambda_\xi}$ is in general discontinous. Fortunately, it is continuous at $\xi = \mu$.
  Instantaneous loss bound:
  - $\sum_{t=1}^n \boldsymbol{E}\Big[\Big(l_t^{\Lambda_\xi}(x_{<t}) - l_t^{\Lambda_\mu}(x_{<t})\Big)^2\Big] \leqslant 2D_n \leqslant 2\ln w_\mu^{-1} < \infty$
  - $0 \leqslant l_t^{\Lambda_\xi}(x_{<t}) - l_t^{\Lambda_\mu}(x_{<t}) \leqslant \sum_{x_t} \Big|\xi(x_t|x_{<t}) - \mu(x_t|x_{<t})\Big| \leqslant \sqrt{2d_t(x_{<t})} \stackrel{t\to\infty}{\longrightarrow} 0$ w.μ.p.1
  - $0 \leqslant l_t^{\Lambda_\xi}(x_{<t}) - l_t^{\Lambda_\mu}(x_{<t}) \leqslant 2d_t(x_{<t}) + 2\sqrt{l_t^{\Lambda_\mu}(x_{<t})d_t(x_{<t})} \underset{w.\mu.p.1}{\stackrel{t\to\infty}{\longrightarrow}} 0$
- The loss function could depend on time and even on individual history, enough that it is bounded: $l_{x_t y_t}^t(x_{<t}) \in [l_{\min}, l_{\max}], l_\Delta := l_{\max} - l_{\min}$.
  Global loss bound:

$$0 \leqslant L_n^{\Lambda_\xi} - L_n^{\Lambda_\mu} \leqslant l_\Delta D_n + \sqrt{4\Big(L_n^{\Lambda_\mu} - n\, l_{\min}\Big) l_\Delta D_n + l_\Delta^2 D_n^2}$$

- **Games of Chance**

  The profit $p_t = -l_{x_t y_t} \in [p_{\max} - p_\Delta, p_{\max}]$, the total profit $P_n^{\Lambda_\rho} = -L_n^{\Lambda_\rho}$ and the average profit per round $\bar{p}_n^{\Lambda_\xi} := \frac{1}{n} P_n^{\Lambda_\xi}$. **Time to win**:

  - $\bar{p}_n^{\Lambda_\xi} = \bar{p}_n^{\Lambda_\mu} - O(n^{-1/2}) \overset{n\to\infty}{\longrightarrow} \bar{p}_n^{\Lambda_\mu}$
  - $n > \left(\frac{2p_\Delta}{\bar{p}_n^{\Lambda_\mu}}\right)^2 k_\mu \wedge \bar{p}_n^{\Lambda_\mu} > 0 \Longrightarrow \bar{p}_n^{\Lambda_\xi} > 0$

  where $w_\mu = e^{-k_\mu}$.

  - **Information-Theoretic Interpretation**: we need that many bits transfered about μ (in the worst case) by received profit. (Read from the book.)

- **Optimality**

  A **prior ξ is Pareto optimal** w.r.t.:

$$s_t, S_n, d_t, D_n, e_t, E_n, l_t, L_n$$

- **Balanced Pareto optimality** w.r.t. L:

$$\Delta_\nu := L_\nu^{\bar{\Lambda}} - L_\nu^{\Lambda_\xi}, \Delta := \sum_{\nu \in \mathcal{M}} w_\nu \Delta_\nu \Longrightarrow \Delta \geqslant 0$$

  In particular, $|\Delta_\eta| \leqslant w_\eta^{-1} \max_{\lambda \in \mathcal{M}} \Delta_\lambda$.

- We have derived bounds for the mean squared sum $S_{n\nu}^{\xi_w} \leqslant \ln w_\nu^{-1}$ and for the loss regret $L_{n\nu}^{\Lambda_{\xi_w}} - L_{n\nu}^{\Lambda_\nu} \leqslant 2 \ln w_\nu^{-1} + 2\sqrt{\ln w_\nu^{-1} L_{n\nu}^{\Lambda_\nu}}$.

  Optimality of universal weights: within the set of enumerable weight functions with short program, the universal weights $w_\nu = 2^{-K(\nu)}$ lead to smallest loss bounds within an additive (to $\ln w_\mu^{-1}$) constant in all enumerable environments.

  - It is difficult to "prove" that universal weights are optimal. See excercise 3.7 in the book.

- Multistep predictions:
  - for horizon h, $\frac{1}{2}\sum \boldsymbol{E}[a^2_{t:n_t}] \leqslant h\, D_\infty \leqslant h \ln w_\mu^{-1}$
  - for arbitrary horizon, convergence in the mean (slow)
- Continuous Probability Classes: **Continuous entropy bound**

$$D_n := \boldsymbol{E}\frac{\mu(x_{1:n})}{\xi(x_{1:n})} \leqslant \ln w_\mu^{-1} + \frac{d}{2}\ln\frac{n}{2\pi} + \frac{1}{2}\ln\det \bar{\jmath}_n + o(1)$$

  where $(\bar{\jmath}_n)^{-1}$ is the Fisher information matrix for the family of distributions $\mathcal{M} = \{\mu_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ and continuity conditions hold (see the book).
- For maximum a posteriori approximator $\rho(x) := \max\{w_\nu \nu(x) : \nu \in \mathcal{M}\}$ or equivalently the minimum description length estimator $\rho(x) := (\lambda x.\mathrm{argmin}_{\nu\in\mathcal{M}}\{\log_2 \nu(x)^{-1} + \log_2 w_\nu^{-1}\})(x)$:

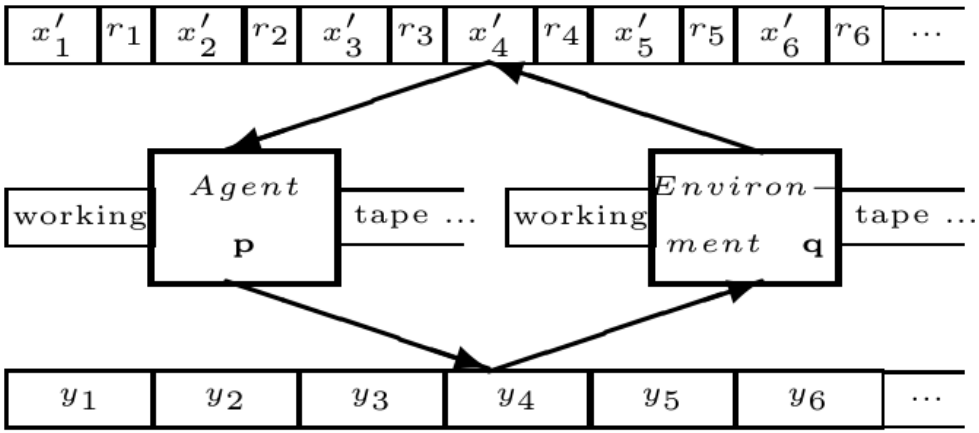
$$\sum_{t=1}^{\infty} \boldsymbol{E}\Big[\sum_{x_t'} \big(\mu(x_t|x_{<t}) - \rho_{(\mathrm{norm})}(x_t'|x_{<t})\big)^2\Big] \overset{\times}{\leqslant} w_\mu^{-1}$$

  where $\rho(x_t|x_{<t}) := \rho(x_{1:t})/\rho(x_{<t})$ and $\rho_{\mathrm{norm}}(x_t|x_{<t}) := \rho(x_{1:t})/\sum_{x_t}\rho(x_{1:t})$ These bounds are tight, thus MDL converges i.m.s. but convergence speed can be exponentially worse than for ξ.

The Universal Algorithmic Agent AIXI

Agents in Known Probabilistic Environments

The agent model (deterministic case):



$p\colon \mathcal{X}^* \to \mathcal{Y}^*$, $y_{1:k} = p(x_{<k})$, $q\colon \mathcal{Y}^* \to \mathcal{X}^*$, $x_{1:k} = q(y_{1:k}) = r_k o_k \in \mathcal{X} := \mathcal{R} \times \mathcal{O}$.

Action y_k is determined by a policy p depending on the I/O history $y_1 x_1 \ldots y_{k-1} x_{k-1} \equiv y x_{<k}$.

Future total reward the agent receives in cycles k to m:

$$V_{km}^{pq} := \sum_{i=k}^{m} r(x_i^{pq})$$

General case: environment μ. The best agent maximizes the expected utility (called value function) $V_\mu^p \equiv V_{1m}^{p\mu} := \sum_q \mu(q) V_{1m}^{pq}$:

$$p^* := \arg\max_p V_{1m}^{pq} \Longrightarrow V_{km}^{p^*q} \geqslant V_{km}^{pq} \;\forall p\colon y_{<k}^{pq} = y_{<k}^{p^*q}$$

AIμ model in Functional Form

- The AIμ model is the agent with policy p^μ that maximizes the μ-expected total reward $r_1 + ... + r_m$, i.e. $p^* \equiv p^\mu := \arg\max_p V_\mu^p$
- In cycle k the (future) value $V_{km}^{p\mu}(yx_{<k})$ of policy p is defined as the μ-expectation of the future reward sum $r_k + ... + r_m$ (the μ, or true, or generating value). Assume in cycle k the history is $\dot{y}\dot{x}_{<k}$, let $\dot{Q}_k := \{q\colon q(\dot{y}_{<k}) = \dot{x}_{<k}\}$ be the set of all environments consistent with this history. Then:

$$V_{km}^{p\mu}(\dot{y}\dot{x}_{<k}) := \frac{\sum_{q\in\dot{Q}_k} \mu(q) V_{km}^{pq}}{\sum_{q\in\dot{Q}_k} \mu(q)}$$

- We generalize the finite lifetime m to a dynamic farsightedness $h_k \equiv m_k - k + 1 \geqslant 1$, called horizon.

$$p_k^* := \arg\max_{p \in \dot{P}_k} V_{k m_k}^{p\mu}(\dot{y}\dot{x}_{<k})$$

where $\dot{P}_k := \{p\colon \exists y_k\colon p(\dot{x}_{<k}) = \dot{y}_{<k} y_k\}$ is the set of policies consistent with the current history.

- By recursively inserting $p_{k-1}^*, \ldots, p_1^*$: (AIμ model)

$$p^*(\dot{x}_{<k}) := p_k^*(\dot{x}_k | p_{k-1}^*(\dot{x}_{<k-1} | \ldots p_1^*)$$

For constant m we have:

$$V_{km}^{*\mu}(\dot{y}\dot{x}_{<k}) \geqslant V_{km}^{p\mu}(\dot{y}\dot{x}_{<k}) \;\forall p \in \dot{P}_k$$

(In sequence prediction it was enough to maximize the next reward, here the sum of future rewards is important.)

AIμ model in Recursive and Iterative Form

Notation: underlined arguments represent probabilistic variables and non-underlined variables represent conditions:

$$\rho(x_{<n}\underline{x}_n) = \frac{\rho(\underline{x}_{1:n})}{\rho(\underline{x}_{<n})}$$

Chronological probability distributions: $\rho(y\underline{x}_{<k}y_k) := \sum_{x_k} \rho(y\underline{x}_{1:k})$. Expected reward:

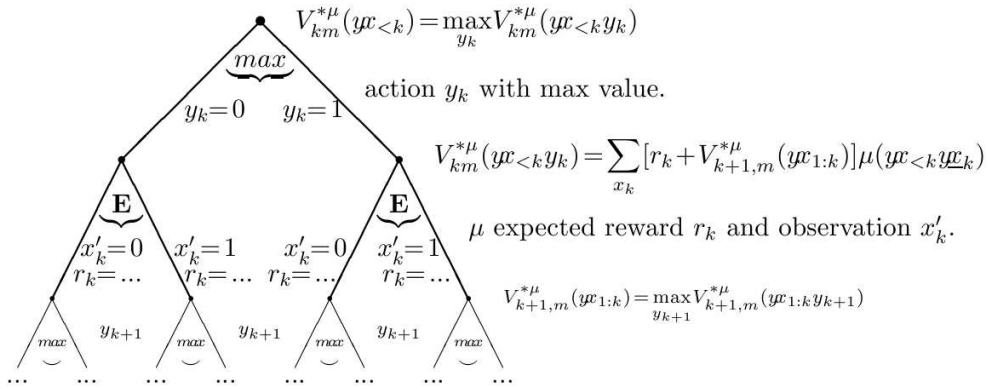
$$V^{*\mu}_{km}(yx_{<k}y_k) := \sum_{x_k} \left[r(x_k) + V^{*\mu}_{k+1,m}(yx_{1:k})\right] \mu(yx_{<k}y\underline{x}_k)$$

How p^* chooses y_k: $V^{*\mu}_{km}(yx_{<k}) := \max_{y_k} V^{*\mu}_{km}(yx_{<k}y_k)$.
Together with the induction start:

$$V^{*\mu}_{m+1,m}(yx_{1:m}) := 0$$

$V^{*\mu}_{km}$ is completely defined:

$$V^{*\mu}_{km}(yx_{<k}) = \max_{y_k} \sum_{x_k} \left[r(x_k) + V^{*\mu}_{k+1,m}(yx_{1:k})\right] \mu(yx_{<k}y\underline{x}_k)$$



If m_k is the horizon function of p^* and $\dot{y}\dot{x}_{<k}$ is the actual history in cycle k,

$$\dot{y}_k = \arg\max_{y_k} V^{*\mu}_{km_k}(\dot{y}\dot{x}_{<k})$$

Unfolding the recursion:

$$\dot{y}_k \equiv \dot{y}^{\mu}_k := \arg\max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} ... \max_{y_{m_k}} \sum_{x_{m_k}} \left(r(x_k) + ... + r(x_{m_k})\right) \mu(\dot{y}\dot{x}_{<k}y\underline{x}_{k:m_k})$$

The value for a general policy p:

$$V^{p\mu}_{km}(yx_{<k}) := \sum_{x_{1:m}} \left(r_k + ... + r_m\right) \mu(yx_{<k}y\underline{x}_{k:m})|_{y_{1:m}=p(x_{<m})}$$

- **Equivalence of Functional and Explicit AI Model**

$$\mu(y\underline{x}_{1:k}) = \sum_{q:\, q(y_{1:k})=x_{1:k}} \mu(q)$$

Factorizable environments $\boldsymbol{\mu}$. Assume that the cycles are grouped into independent episodes:

$$\mu(y\underline{x}_{1:n}) = \prod_{r=0}^{s-1} \mu_r(y\underline{x}_{n_r+1:n_{r+1}})$$

Then $\dot{y}_k$ depends on μ_r and x and y of episode r only:

$$\dot{y}_k = \arg\max_{y_k} \sum_{x_t} \left(r(x_k) + \ldots + r(x_t)\right) \mu_r(\dot{y}\dot{x}_{n_r+1:k-1} y\underline{x}_{k:t})$$

with $t := \min\{m_k, n_{r+1}\}$.

Probabilistic policies. For policy π:

$$\begin{aligned} V_\mu^\pi &= \sum_{yx_{1:m}} \Big((r_1 + \ldots + r_m)\, \mu(x_m | yx_{<m} y_m)\, \pi(y_m | yx_{<m}) \\ &\qquad \ldots \mu(x_1|y_1)\pi(y_1)) \\ &= \sum_{yx_{1:m}} (r_1 + \ldots + r_m) \mu(x_{1:m}|y_{1:m})\, \pi(y_{1:m}|x_{<m}) \\ &\equiv \sum_{yx_{1:m}} (r_1 + \ldots + r_m) \rho(y_{1:m} x_{1:m}) \end{aligned}$$

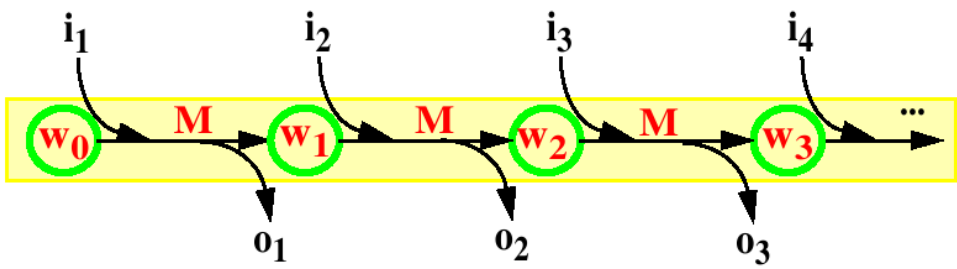
Among optimal policies there is always a deterministic one:

$$\max_\pi V_\mu^\pi = \max_p V_\mu^p$$

Persistent Turing Machines ⟨reference|PTMs⟩

- independently introduced model of interactive computation based on non-deterministic monotonic Turing machines (three tapes: read-only, work, write-only)
- "cuts the environment out of the loop"
  - inputs are arbitrary
  - based on coinductive notions (coalgebras, LTSs, bisimulation)
- stresses infinite input / output alphabets (e.g. strings)

PTM operation:



The Universal Algorithmic Agent AIXI

Replace the unknown prior probability μ^{AI} in the AIμ model by a universal prior semi-probability M^{AI} with $M(q) := 2^{-l(q)}$. Equivalence of functional and iterative AI model still holds, equivalence with recursive AI model holds after normalization of M (which will no longer be enumerable, but the universal value $V_{km}^{p\xi}$ will still be enumerable). q knows length of the output from input length.

$$M(y\underline{x}_{1:k}) = \sum_{q:\, q(y_{1:k})=x_{1:k}} 2^{-l(q)}$$

Summing over all enumerable chronological semimeasures:

$$\xi(y\underline{x}_{1:n}) := \sum_{\rho} 2^{-K(\rho)} \rho(y\underline{x}_{1:n}) \stackrel{\times}{=} M(y\underline{x}_{1:n})$$

Just like in the sequence prediction case:

$$\sum_{k=1}^{n} \sum_{x_{1:k}} \mu(y\underline{x}_k) \Big(\mu(yx_{<k}y\underline{x}_k) - \xi(yx_{<k}y\underline{x}_k) \Big)^{2+} \leqslant \ln 2\, K(\mu)$$

and

$$\xi(yx_{<k}y\underline{x}'_{k:m_k}) \stackrel{k\to\infty}{\longrightarrow} \mu(yx_{<k}y\underline{x}'_{k:m_k}) \begin{cases} i.m.s. \text{ if } h_k = m_k - k + 1 \leqslant h_{\max} < \infty \\ i.m. \text{ for general } m_k \end{cases}$$

Intelligence Order Relation. Extend the ξ-expected reward definition to programs that are not consistent with the current history:

$$V_{km}^{p\xi}(\dot{y}\dot{x}_{<k}) := \frac{1}{\mathcal{N}} \sum_{q:\, q(\dot{y}_{<k})=\dot{x}_{<k}} 2^{-l(q)}\, V_{km}^{\tilde{p}q}$$

where $\mathcal{N}$ is the normalization factor only necessary for the expectation interpretation. For $p \notin \dot{P}_k$, $\tilde{p}$ is p modified to "output" $\dot{y}\dot{x}_{<k}$ but unaltered for further cycles.

- p is more or equally intelligent than p' iff

$$p \succcurlyeq p' \equiv \forall k \forall \dot{y}\dot{x}_{<k}\colon V_{km_k}^{p\xi}(\dot{y}\dot{x}_{<k}) \geqslant V_{km_k}^{p'\xi}(\dot{y}\dot{x}_{<k})$$

For completely unknown μ we could take $\xi = M$ and treat AIXI as optimal by construction (similarily to when we take uniform prior over parameters for the bandits problem).

Separability concepts.

- Self-optimizing policies: for p^{best} independent of μ

$$V_{1m}^{p^{\text{best}}\mu} \geqslant V_{1m}^{p\mu} - o(\ldots) \quad \forall \mu, p$$

alternatively, $\frac{1}{m} V_{1m}^{p^{\text{best}}\mu} \stackrel{m\to\infty}{\longrightarrow} \frac{1}{m} V_{1m}^{p\mu}$.

- The HeavenHell example:
- The OnlyOne example:

$$\mathcal{M} := \Big\{ \mu_{y^*} \colon y^* \in \mathcal{Y}, K(y^*) = \lfloor \log_2 |\mathcal{Y}| \rfloor, \mu_{y^*}(y x_{<k} y_k \underline{1}) := \delta_{y_k y^*} \forall k \Big\}$$

There are $N \stackrel{\times}{=} |\mathcal{Y}|$ such y^*. The number of errors is

$$E_\infty^p \geqslant N - 1 \stackrel{\times}{=} |\mathcal{Y}| \stackrel{\times}{=} 2^{K(y^*)} \stackrel{\times}{=} 2^{K(\mu)}$$

$2^{K(\mu)}$ is the best possible bound depending on $K(\mu)$; it could be OK if $K(\mu|\dot{x}_{<k}) = O(1)$.

- μ is passive if environment is not influenced by agent output. $\mathcal{M}$ and $\mu \in \mathcal{M}$ are pseudo-passive if the corresponding $p^{\text{best}} = p^\xi$ is self-optimizing.

- The μ-expected number of suboptimal choices

$$D_{n\mu\xi} := \boldsymbol{E}\Bigg[\sum_{k=1}^{n} 1 - \delta_{\dot{y}_k^\mu, \dot{y}_k^\xi}\Bigg], \text{ where } \dot{y}_k^\mu \equiv p^\mu(\dot{y}\dot{x}_{<k})$$

  μ can be asymptotically learned if

$$D_{n\mu\xi}/n \overset{n\to\infty}{\longrightarrow} 0, \text{ i.e. } D_{n\mu\xi} = o(n)$$

  Claim: AIXI can asymptotically learn any relevant problem.

- μ is uniform if

$$\Big|\mu(yx_{<k}y'_k\underline{x}'_k) - \xi(yx_{<k}y'_k\underline{x}'_k)\Big| < c\,\Big|\mu(yx_{<k}y_k\underline{x}_k) - \xi(yx_{<k}y_k\underline{x}_k)\Big| \;\forall y'_k x'_k$$

  There are relevant μ that are not uniform. Uniform μ can be asymptotically learned for appropriately weighted $D_{n\mu\xi}$ and bounded horizon.

- μ is forgetful if $\mu(yx_{<k}y\underline{x}_k)$ becomes independent of $yx_{<l}$ for fixed l and $k\to\infty$. μ is farsighted if $\lim_{m_k\to\infty} \dot{y}_k^{(m_k)}$ exists.

- Markovian, generalized (l^{th} order) Markovian, ergodic, factorizable.

Value-Related Optimality, Discounted Future Value Function

The γ discounted weighted-average future value of probabilistic policy π in environment ρ given history $yx_{<k}$ (the ρ-value of π given $yx_{<k}$):

$$V_{k\gamma}^{\pi\rho}(yx_{<k}) := \frac{1}{\Gamma_k} \lim_{m\to\infty} \sum_{yx_{k:m}} (\gamma_k r_k + \ldots + \gamma_m r_m)\rho(yx_{<k}y\underline{x}_{k:m})\pi(yx_{<k}\underline{y}x_{k:m})$$

with $\Gamma_k := \sum_{i=k}^{\infty} \gamma_i$. The discounted AI$\rho$ model is defined as the policy p^ρ:

$$p^\rho := \arg\max_\pi V_{k\gamma}^{\pi\rho},\ V_{k\gamma}^{*\rho} := V_{k\gamma}^{p^\rho\rho} = \max_\pi V_{k\gamma}^{\pi\rho} \geqslant V_{k\gamma}^{\pi\rho}\ \forall\pi$$

- **Linearity** and **convexity** of V_ρ in ρ:

$$V_{k\gamma}^{\pi\xi} = \sum_{\nu\in\mathcal{M}} w_k^\nu V_{k\gamma}^{\pi\nu} \quad \text{and} \quad V_{k\gamma}^{*\xi} \leqslant \sum_{\nu\in\mathcal{M}} w_k^\nu V_{k\gamma}^{*\nu}$$

  where $\xi(yx_{<k}y\underline{x}_{k:m}) = \sum_{\nu\in\mathcal{M}} w_k^\nu \nu(yx_{<k}y\underline{x}_{k:m})$ with $w_k^\nu := w_\nu \frac{\nu(y\underline{x}_{<k})}{\xi(y\underline{x}_{<k})}$.

- **Pareto optimality**: there is no other policy π with $V_{k\gamma}^{\pi\nu} \geqslant V_{k\gamma}^{p^\xi\nu}\ \forall\nu\in\mathcal{M}$ and strict inequality for at least one ν.

- Balanced Pareto optimality:

$$0 \leqslant V_{k\gamma}^{*\nu} - V_{k\gamma}^{\tilde{\pi}_k \nu} =: \Delta_k^\nu \implies 0 \leqslant V_{k\gamma}^{*\nu} - V_{k\gamma}^{p^\xi \nu} \leqslant \frac{1}{w_k^\nu} \Delta_k$$

  with $\Delta_k := \sum_{\nu \in \mathcal{M}} w_k^\nu \Delta_k^\nu$, where all quantities depend on history $yx_{<k}$.

- If there exists a sequence of self-optimizing policies $\tilde{\pi}_k$ **then the universal policy p^ξ is self-optimizing**

$$\exists \tilde{\pi}_k \forall \nu \colon V_{k\gamma}^{\tilde{\pi}_k \nu} \stackrel{k \to \infty}{\longrightarrow} V_{k\gamma}^{*\nu} \quad \text{w.}\nu\text{.p.1} \quad \implies V_{k\gamma}^{p^\xi \mu} \stackrel{k \to \infty}{\longrightarrow} V_{k\gamma}^{*\mu} \quad \text{w.}\mu\text{.p.1}$$

  where the probabilities are conditional on historic perceptions $x_{<k}$.

- The values $V_{k\gamma}^{\pi\mu}$ and $V_{k\gamma}^{*\mu}$ are **continuous** in μ, and $V_{k\gamma}^{p^{\hat{\mu}}\mu}$ is **continuous** in $\hat{\mu}$ **at** $\hat{\mu} = \mu$: If

$$\sum_{x_k} \left| \mu(yx_{<k}y\underline{x}_k) - \mu(yx_{<k}y\underline{x}_k) \right| \leqslant \varepsilon \forall yx_{<k}y_k \forall k \geqslant k_0$$

then

i. $\left| V_{k\gamma}^{\pi\mu} - V_{k\gamma}^{\pi\hat{\mu}} \right| \leqslant \delta(\varepsilon)$

ii. $\left| V_{k\gamma}^{*\mu} - V_{k\gamma}^{*\hat{\mu}} \right| \leqslant \delta(\varepsilon)$

iii. $\left| V_{k\gamma}^{*\mu} - V_{k\gamma}^{p^{\hat{\mu}}\mu} \right| \leqslant 2\delta(\varepsilon)$

for all $k \geqslant k_0$ and $yx_{<k}$, where $\delta(\varepsilon) = r_{\max} \min_{n \geqslant k} \left\{ (n-k)\varepsilon + \frac{\Gamma_n}{\Gamma_k} \right\} \xrightarrow{\varepsilon \to 0} 0.$

- If $y_{1:m} = p(x_{1:m})$ (on-policy) and $V^{p\cdots}_{k\cdots} = V^{p\cdots}_{k\cdots}(yx_{<k})$, then the **universal** undiscounted future **value** $V^{p\xi}_{km_k}$ with bounded dynamic horizon $h_k = m_k - k + 1$ **converges** i.m.s. **to the true value** $V^{p\mu}_{km_k}$ and the discounted future value $V^{p\xi}_{k\gamma}$ converges i.m. to $V^{p\mu}_{k\gamma}$ for any summable discount sequence γ_k.

  i. $\left|V^{p\xi}_{km} - V^{p\mu}_{km}\right| \leqslant (m-k+1) r_{\max} a_{k:m}$

  $\left|V^{p\xi}_{k\gamma} - V^{p\mu}_{k\gamma}\right| \leqslant r_{\max}\sqrt{2d_{k:\infty}}$

  ii. $\sum_{k=1}^{\infty} \boldsymbol{E}\Big(V^{p\xi}_{km_k} - V^{p\mu}_{km_k}\Big)^2 \leqslant 2h^3_{\max} r^2_{\max} D_\infty$

  $\boldsymbol{E}\Big(V^{p\xi}_{k\gamma} - V^{p\mu}_{k\gamma}\Big)^2 \leqslant 2r^2_{\max}(D_\infty - D_{k-1}) \stackrel{k\to\infty}{\longrightarrow} 0$

  iii. $V^{p\xi}_{km_k} \stackrel{k\to\infty}{\longrightarrow} V^{p\mu}_{km_k}$ i.m.s. if $h_{\max} < \infty$

  $V^{p\xi}_{k\gamma} \stackrel{k\to\infty}{\longrightarrow} V^{p\mu}_{k\gamma}$ i.m. for any γ

where $a_{k:m} := \sum_{x_{k:m}} \Big|\mu(yx_{<k}y\underline{x}_{k:m}) - \xi(yx_{<k}y\underline{x}_{k:m})\Big|$,

$d_{k:m} := \sum_{x_{k:m}} \mu(yx_{<k}y\underline{x}_{k:m}) \ln \frac{\mu(yx_{<k}y\underline{x}_{k:m})}{\xi(yx_{<k}y\underline{x}_{k:m})}$, $D_k := d_{1:k} \leqslant \ln w_\mu^{-1} < \infty$.

- Markov decision process is ergodic, if there exists a policy which visits each state infinitely often with probability 1. There exist **self-optimizing policies** $\tilde{p}_m$ for the class of **ergodic MDPs**:

$$\exists \tilde{p}_m \forall \nu \in \mathcal{M}_{\text{MDP1}}: \frac{1}{m} V_{1m}^{*\nu} - \frac{1}{m} V_{1m}^{\tilde{p}_m \nu} = O(m^{-1/3})$$

With unbounded effective horizon $h_k^{\text{eff}} \stackrel{k\to\infty}{\longrightarrow} \infty$:

$$\exists \tilde{\pi}_k \forall \nu \in \mathcal{M}_{\text{MDP1}}: \frac{1}{m} V_{1m}^{\tilde{\pi}_k \nu} \stackrel{k\to\infty}{\longrightarrow} \frac{1}{m} V_{1m}^{*\nu} \text{ for any history } yx_{<k} \text{ if } \frac{\gamma_{k+1}}{\gamma_k} \to 1.$$

If $\mathcal{M}$ is a countable clas of ergodic MDPs, $\xi := \sum_{\nu\in\mathcal{M}} w_\nu \nu$, then $\text{AI}\xi \equiv p_m^\xi$ maximizing $V_{1m}^{p\xi}$ and p^ξ maximizing $V_{k\gamma}^{\pi\xi}$ are **self-optimizing**:

$$\forall \nu \in \mathcal{M}: \frac{1}{m} V_{1m}^{p_m^\xi \nu} \stackrel{m\to\infty}{\longrightarrow} \frac{1}{m} V_{1m}^{*\nu} \quad \text{and} \quad V_{k\gamma}^{p_m^\xi \nu} \stackrel{k\to\infty}{\longrightarrow} V_{k\gamma}^{*\nu} \quad \text{if} \quad \frac{\gamma_{k+1}}{\gamma_k} \to 1$$

If $\mathcal{M}$ is finite, then the speed of first convergence is at least $O(m^{-1/3})$.

- Ergodic POMDPs, ergodic l^{th}-order MDPs and factorizable environments also allow self-optimizing policies.

The Choice of the Horizon.

- Fixed (effective) horizon is OK if we know the lifetime of the agent, e.g. if the probability of surviving to the next cycle is (always, independently) $\gamma < 1$, we can choose geometric discount rate γ.

- General discounting introduces unbounded effective horizons.

  Let $r_k := \gamma_k r'_k$ with $\gamma_k > 0$ and $r'_k \in [0, 1]$. If $\Gamma_k := \sum_{i=k}^{\infty} \gamma_i < \infty$, then $V_{k\gamma}^{p\rho} := \frac{1}{\Gamma_k} \lim_{m\to\infty} V_{km}^{p\rho}$ exists. β-effective horizon $h_k^\beta := \min\{h \geqslant 0 \colon \Gamma_{k+h} \leqslant \beta\, \Gamma_k\}$. Approximating $V_{k\gamma}$ by the first h_k^β terms introduces an error of at most $\beta r_{\max}$. $h_k^{\text{eff}} := h_k^{\beta=1/2}$.

Horizons	γ_k	$\Gamma_k = \sum_{i=k}^{\infty} \gamma_i$	h_k^β
finite	$\begin{cases} 1 \text{ for } k \leqslant m \\ 0 \text{ for } k > m \end{cases}$	$m - k + 1$	$\lceil (1-\beta)(m-k+1) \rceil$
geometric	$\gamma^k, 0 \leqslant \gamma < 1$	$\frac{\gamma^k}{1-\gamma}$	$\left\lceil \frac{\ln \beta}{\ln \gamma} \right\rceil$
power	$k^{-1-\varepsilon}, \varepsilon > 0$	$\sim \frac{1}{\varepsilon} k^{-\varepsilon}$	$\sim (\beta^{-1/\varepsilon} - 1) k$
harmonic	$\frac{1}{k \ln^{1+\varepsilon} k}, \varepsilon > 0$	$\sim \frac{1}{\varepsilon} (\ln k)^{-\varepsilon}$	$\sim k^{\beta - 1/\varepsilon}$
universal	$2^{-K(k)}$	decreases slower than any computable function	increases faster than any computable function

- Infinite horizon: take $\dot{y}_k^{(\infty)} \in \liminf_{m\to\infty} \dot{\mathcal{Y}}_k^{(m)}$, where $\dot{\mathcal{Y}}_k^{(m)} := \Big\{ \dot{y}_k^{(m_k)} : m_k \geqslant m \Big\}$. Limit $\lim_{m\to\infty} V^*_{km}(yx_{<k})$ needs not exist. But **immortal agents are lazy**: if postponing reward makes it bigger, the agent construed this way will not get any reward.
- For $\rho \notin \mathcal{M}$, let $\xi' := (1-\alpha)\xi + \alpha\rho$, then

$$\limsup_{k\to\infty} \Big[V^{*\mu}_{k\gamma} - V^{p^{\xi'}\mu}_{k\gamma} \Big] \leqslant \frac{\alpha\, r_{\max}}{(1-\alpha) w_\mu}$$

  with examples where equality holds. Thus a belief contamination of magnitude α comparable to w_μ can completely degenerate performance. (???)
- (Posterization) It is not true, that if $w_\nu = 2^{-K(\nu)}$, then $w_k^\nu \sim 2^{-K(\nu|yx_{<k})}$, for $\sum_{\nu\in\mathcal{M}} w_k^\nu \nu(.|yx_{<k}) := \xi(.|yx_{<k})$.

Actions as random variables

Instead of defining ξ as a mixture of environments, we could use a universal distribution over perceptions and actions and then conditionalize to the actions:

$$\xi^{\mathrm{AI}}_{\mathrm{alt}}(y\underline{x}_{1:n}) := \frac{M(\underline{y}x_{1:n})}{\sum_{x_{1:n}} M(\underline{y}x_{1:n})}$$

where M is the Solomonoff's prior (we could use ξ_U as well). Open problems:

- is $\xi^{\mathrm{AI}}_{\mathrm{alt}}$ enumerable?
- $\xi^{\mathrm{AI}}_{\mathrm{alt}} \stackrel{\times}{=} \xi^{\mathrm{AI}}$?
- could $M(yx_{<k}\underline{y}_k)$ be close to the action of p^{ξ} and/or $p^{\xi_{\mathrm{alt}}}$ for large k, justifying the interpretation that $M(yx_{<k}\ \underline{y}_k)$ is the agent's own belief in selecting action y_k?

Uniform mixture of MDPs

Let $\mu_T \in \mathcal{M}_{\text{MDP}}$ be a completely observable MDP with transition matrix T.

$$\mu_T(a_1 \underline{s}_1 \ldots a_n \underline{s}_n) = T^{a_1}_{s_0 s_1} \cdot \ldots \cdot T^{a_n}_{s_{n-1} s_n}$$

Reward is a function of state $r_k = r(s_k)$. The Bayes mixture

$$\xi(a \underline{s}_{1:n}) := \int_{\mathcal{T}} w_T \mu_T(a \underline{s}_{1:n}) dT$$

- For a uniform prior belief,

$$\xi(a\, s_{<n} a \underline{s}_n) = \frac{\xi(a \underline{s}_{1:n})}{\xi(a \underline{s}_{<n})} = \frac{N^{a_n}_{s_{n-1} s_n}}{\sum_{s'} N^{a_n}_{s_{n-1} s'} + S - 1}$$

  where $S = |\mathcal{X}|$ is the number of states and $N^a_{s\,s'}$ is the historical (i.e. in $a\, s_{1:n}$) number of transitions from s to s' under a.

- Although $\mathcal{T}$ is continuous and contains non-ergodic environments, the Bayes optimal policy p^ξ is self-optimizing for ergodic environments $\mu_T \in \mathcal{M}_{\text{MDP1}}$. (Intuition: $\mathcal{T}$ is compact and non-ergodic environments have measure zero.)

- Posterior belief $w_T(a\,s_{1:n}) \propto \mu_T(a\,\underline{s})$ is a (complex) distribution over possible T. Most RL algorithms only estimate a single T (e.g. a most likely, or an expected T). Policy p^ξ appropriately explores the environment, while the popular policy based on $\boldsymbol{E}[T]$, or on Maximum Likelihood, lack exploration. Expected transition probability:

$$\boldsymbol{E}[T^a_{s\,s'}|a\,s_{1:n}] := \int_{\mathcal{T}} T^a_{s\,s'} w_T(a\,s_{1:n}) dT = \frac{N^a_{s\,s'}+1}{\sum_{s'} N^a_{s\,s'}+S}$$

Important Environmental Classes

Here $\xi = \xi_U \stackrel{\times}{=} M$ is the Solomonoff's prior, i.e. $\mathrm{AI}\xi = \mathrm{AIXI}$.

Sequence Prediction

The $\mathrm{AI}\mu$ model and $\mathrm{SP}\mu = \Theta_\mu$ model (derived for binary alphabet $\mathbb{B}$) for known environment μ are equivalent and the expected prediction error relates to the value function:

$$V_{1m}^{*\mu} = m - E_m^{\Theta_\mu}$$

The general ξ is not symmetric in $y_i r_i \leftrightarrow (1-y_i)(1-r_i)$ thus more difficult. We concentrate on deterministic computable envirnment $\dot{z} = \dot{z}_1 \dot{z}_2 ...$ with $Km(\dot{z}_1 ... \dot{z}_n) \leqslant Km(\dot{z}) < \infty$ and horizon $m_k = k$ (greedily maximize next reward; this is sufficient for SP but does not show the behavior of AIXI for a universal horizon). We have (best proven bound):

$$E_\infty^{\mathrm{AI}\xi} < \frac{1}{\alpha} = 2^{Km(\dot{z})+O(1)}$$

The intuitive interpretation is that each wrong prediction eliminates at least one program p of size $l(p) \stackrel{+}{\leqslant} Km(\dot{z})$. The best possible bound is:

$$E_\infty^{\mathrm{SP}\xi} \stackrel{+}{\leqslant} 2\ln 2 \, Km(\dot{z})$$

Strategic Games

We restrict ourselves to deterministic strictly competitive strategic games. Assume a bounded length game padded to length n. Assume the environment uses the minimax strategy:

$$\dot{o}_k = \arg\min_{o_k} \max_{y_{k+1}} \min_{o_{k+1}} \ldots \max_{y_n} \min_{o_n} V(\dot{y}_1\dot{o}_1 \ldots \dot{y}_k o_k \ldots y_n o_n)$$

$r_1 = \ldots = r_{n-1} = 0$, $r_n = 1$ if the AIμ agent wins, $\frac{1}{2}$ for draw, and 0 if environment wins. Illegal move is an instant loss.

$$\begin{aligned}
\dot{y}_k^{\mathrm{AI}} &= \arg\max_{y_k} \sum_{o_k} \ldots \max_{y_n} \sum_{o_n} V(\dot{y}\dot{o}_{<k} y o_{k:n})\, \mu^{\mathrm{SG}}(\dot{y}\dot{o}_{<k} \underline{y}\underline{o}_{k:n}) \\
&= \arg\max_{y_k} \sum_{o_k} \ldots \max_{y_{n-1}} \sum_{o_{n-1}} \max_{y_n} \min_{o_n} V(\dot{y}\dot{o}_{<k} y o_{k:n})\, \mu^{\mathrm{SG}}(\dot{y}\dot{o}_{<k} \underline{y}\underline{o}_{k:n-1}) \\
&= \ldots = \arg\max_{y_k} \min_{o_{k+1}} \ldots \max_{y_n} \min_{o_n} V(\dot{y}\dot{o}_{<k} y o_{k:n}) = \dot{y}_k^{\mathrm{SG}}
\end{aligned}$$

If the game is played multiple times, then μ is factorizable. But if the game has variable length, then μ is no longer factorizable: a better player prefers short games and prefers a quick draw over too long a win.

Using the AIξ model for game playing. The AIξ agent has only much less than $\log_2 3$ information from a single game, so needs at least $O(K(\text{game}))$ number of games. Variable length games are better: the AIξ agent will quickly learn legal moves from short initial games. Next, AIξ will learn the losing positions. Then, AIξ will win some games by luck or will exploit the game symmetry to learn winning positions.
The AIξ agent can take advantage of environmental players with limited rationality by not settling on the minimax strategy.

Function Minimization

We will consider distributions over functions:

$$\mu^{\mathrm{FM}}(y_1 z_1 \ldots y_n z_n) := \sum_{f: f(y_i)=z_i, \forall 1 \leqslant i \leqslant n} \mu(f)$$

Greedy model is not appropriate, because it will stick with an argument of value already below the expectations for other values. For episodes of length m:

$$\dot{y}_k^{\mathrm{FM}} = \arg\min_{y_k} \sum_{z_k} \ldots \min_{y_m} \sum_{z_m} (\alpha_1 z_1 + \ldots + \alpha_m z_m)\, \mu(\dot{y}_1 \dot{z}_1 \ldots \dot{y}_{k-1} \dot{z}_{k-1} y_k z_k \ldots y_m z_m)$$

If we want the last output to be optimal, set $\alpha_1 = \ldots = \alpha_{m-1} = 0, \alpha_m = 1$ (FMFξ); if we want already good approximation along the way, set $\alpha_1 = \ldots = \alpha_m = 1$ (FMSξ), etc.
For the FMξ model:

$$\xi^{\mathrm{FM}}(y_1 z_1 \ldots y_n z_n) := \sum_{q: q(y_i)=z_i, \forall 1 \leqslant i \leqslant n} 2^{-l(q)}$$

FMξ will never cease searching for minima, will test an infinite set of y's for $m \to \infty$. FMFξ will never repeat any y except at $t = m$. FMSξ will test a new y_t (for fixed m) only if the expected $f(y_t)$ is not too large.

For AIμ/ξ, we need $r_k = -\alpha_k z_k, o_k = z_k$.
AIξ has problem with the FMF model: it must first learn that it has to minimize a function. It can learn it by repeated minimization of (different) functions.

Supervised Learning from Examples (EX)

The environment presents inputs $o_{k-1} = z_k v_k \equiv (z_k, v_k) \in R \cup (\mathcal{Z} \times \{?\}) \subset \mathcal{Z} \times (\mathcal{Y} \cup \{?\}) = \mathcal{O}$. The relations R might be probability distributed with $\sigma(\underline{R})$.

$$\mu^{\mathrm{AI}}(y_1 \underline{x}_1 \ldots y_n \underline{x}_n) = \sum_{R:\forall 1<i\leqslant n\, R(z_i, y_i)=r_i} \mu_R(\underline{o_1 \ldots o_n})\, \sigma(\underline{R})$$

where $x_i = r_i o_i$ and $o_{i-1} = z_i v_i$ with $v_i \in \mathcal{Y} \cup \{?\}$. The AI$\xi$ agent only needs $O(1)$ bits from reinforcement r_k to learn to extract z_i from o_{i-1} and return y_i with $(z_i, y_i) \in R$.

AIXItl and Optimal Search

The Fastest and Shortest Algorithm for All Problems

Let

- p^*: a given algorithm or a specification of a function
- p: any prog. provably the same as p^* with time complexity provably in t_p
- $\text{time}_{t_p}(x)$: the time needed to compute $t_p(x)$

For fixed $\varepsilon \subset (0, \frac{1}{2})$ the algorithm $M^{\varepsilon}_{p^*}$ computes $p^*(x)$ in time

$$\boxed{\text{time}_{M^{\varepsilon}_{p^*}}(x) \leqslant (1+\varepsilon)\, t_p(x) + \frac{d_p}{\varepsilon}\, \text{time}_{t_p}(x) + \frac{c_p}{\varepsilon}}$$

with constants c_p and d_p independent of x.
For example, if matrix multiplication algorithm p'' with $\text{time}_{p''}(x) = d\ n^2 \log\ n$ exists, then $\text{time}_{M^{\varepsilon}_{p^*}}(x) \leqslant (1+\varepsilon)\, d\, n^2 \log n + O(1)$ for all x.
The Blum's speed-up theorem doesn't affect this, because its speed-up sequence is not computable.
Always $\boxed{\text{time}_{t_p(x)} \leqslant t_p(x)}$ and the multiplicative constant $1 + \varepsilon + d_p$ is huge if no good time bound exists. This could be the case for many complex approximation problems and for universal reinforcement learning.

Levin Search

Inverses a quickly computable function g devoting $2^{-l(p)}$ time portion to the potential inverse algorithm p. Computation time: $2^{l(p)}$ $\text{time}_p^+(x)$ ($\text{time}_p^+(x)$ for checking $g(p(x)) = x$).

It can be implemented as Li & Vitanyi's SEARCH(g): run every p shorter than i for $2^i 2^{-l(p)}$ steps in phase $i = 1, 2, 3, \ldots$ until g is inverted on x.

The Fast Algorithm $M^{\varepsilon}_{p^*}$

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

Initialize the shared variables $L := \{\}, t_{\text{fast}} := \infty, p_{\text{fast}} := p^*$.
Start algorithms A, B, C in parallel with relative computational resources $\varepsilon, \varepsilon, 1-2\varepsilon$ respectively.

Algorithm A.

Systematically search the space of proofs for proofs of formulas of the form

$$\forall y\colon u(?p, y) = u(p^*, y) \wedge u(?t, y) \geqslant \text{time}(?p, y)$$

For each such proof, add the answer substitution (p, t) to L.

Algorithm B. `for all` $(p, t) \in L$:

Run U on (t, x) in parallel for all t with relative computational resources $2^{-l(p)-l(t)}$.
`if` U halts for some t and $U(t, x) < t_{\text{fast}}$
`then` $t_{\text{fast}} := U(t, x)$ and $p_{\text{fast}} := p$ and restart algorithm C.

Algorithm C.

Run U on (p_{fast}, x). For each executed step decrease t_{fast} by 1.
`if` U halts `then` abort computation of A and B and return $U(p_{\text{fast}}, x)$.

Time-Bounded AIXI Model

Time-Limited Probability Distributions.
We could limit environments in the universal mixture ξ to those of length $\tilde{l}$ computable in time $\tilde{t}$ arriving at expectimax algorithm with complexity $t(\dot{y}_k^{\mathrm{AI}\xi^{\tilde{t}\tilde{l}}}) = O(|\mathcal{Y}|^{h_k}\,|\mathcal{X}|^{h_k}\,2^{\tilde{l}}\,\tilde{t})$. (Considered poor – unintelligent.)

Best Vote Algorithm.
Without normalization, the ξ-expected future reward is enumerable:

$$V_{km}^{p\xi}(\dot{y}\dot{x}_{<k}) := \sum_{q\in\dot{Q}_k} 2^{-l(q)} V_{km}^{pq},\ V_{km}^{pq} := r(x_k^{pq}) + \ldots + r(x_m^{pq})$$

At every cycle we select the best (possibly inconsistent with history) policy. But V_{km_k} is uncomputable (approximable, the same effort as computing $\dot{y}_k^{\mathrm{AI}\xi}$): let the policy estimate it (by w_k^p):

$$p(\dot{y}\dot{x}_{<k}) = w_1^p y_1^p \ldots w_k^p y_k^p$$

No policy allowed to claim to be better than it is: valid approximation

$$\mathrm{VA}(p) \equiv [\forall k \forall w_1^p y_1^p \dot{y}_1 \dot{x}_1 \ldots w_k^p y_k^p\colon p(\dot{y}\dot{x}_{<k}) = w_1^p y_1^p \ldots w_k^p y_k^p \Longrightarrow w_k^p \leqslant V_{km_k}^{p\xi}(\dot{y}\dot{x}_{<k})$$

$V_{km_k}^{*\xi}$ is enumerable: $\exists(p_i)\ \mathrm{VA}(p_i) \wedge \lim_{i\to\infty} w_k^{p_i} = V_{km_k}^{*\xi}$. The convergence is not uniform in k, but it is OK: we select a policy in each step.

The Universal Time-Bounded AIXItl Agent.

1. Systematically search the space of proofs shorter than l_P for VA($?p$) and collect all answer substitutions.
2. Eliminate all p of length $>\tilde{l}$.
3. Modify each p to stop within $\tilde{t}$ time steps (aborting with $w_k=0$ if needed)
4. Start first cycle: $k:=1$.
5. Run every $p(\dot{y}\dot{x}_{<k}) = w_1^p y_1^p ... w_k^p y_k^p$, where all outputs are redirected to some auxiliary tape, incrementally by adding $\dot{y}\dot{x}_{k-1}$ to the input tape and continuing the computation of the previous cycle.
6. Select $p_k^* := \arg\max_p w_k^p$.
7. Write $\dot{y}_k := y_k^{p_k^*}$ to the output tape.
8. Receive input $\dot{x}_k$ from the environment.
9. Begin next cycle $k:=k+1$, goto step 5.

[Compare with e.g. accuracy-based learning classifier systems XCS by Wilson, market/economy RL by Baum & Durdanovic.]

- We call p effectively more or equally intelligent than p' if

$$\begin{aligned} p \succcurlyeq^c p' \;\equiv\; & \forall k \forall \dot{y}\dot{x}_{<k} \exists w_{1:n} w'_{1:n}: \\ & p(\dot{y}\dot{x}_{<k}) = w_1 * \ldots w_k * \\ & \wedge\, p(\dot{y}\dot{x}_{<k}) = w_1 * \ldots w_k * \wedge w_k \geqslant w'_k \end{aligned}$$

- **Optimality of AIXItl.** Let p be any extended chronological (incremental) program (like above) of length $l(p) \leqslant \tilde{l}$ and computation time per cycle $t(p) \leqslant \tilde{t}$, for which there exists a proof of $\mathrm{VA}(p)$ of length $\leqslant l_P$, then $p^* \succcurlyeq^c p$. The length of p^* is $\boxed{l(p^*) = O(\log(\tilde{l}\,\tilde{t}\,l_P))}$, the setup time is (at most, depending on the proof search technique) $\boxed{t_{\mathrm{setup}}(p^*) = O(l_P^2\, 2^{l_P})}$ and the computation time per cycle is $\boxed{t_{\mathrm{cycle}}(p^*) = O(2^{\tilde{l}}\,\tilde{t})}$ (to go faster, we could eliminate provably poor policies: how many Pareto-good policies are there?)

- There could be policies which produce good outputs within reasonable time, but their justification w^p or proof $\mathrm{VA}(p)$ take unreasonably long.

- The inconsistent programs p must be able to continue strategies started by other policies. A policy can steer the environment to a direction for which it is specialized. This requires enough separability to "recover".

Algorithm/Properties	time efficient	data efficient	explo-ration	conver-gence	global optimum	genera-lization	POMDP	learning	active
Value/Policy iteration	yes/no	yes	–	YES	YES	NO	NO	NO	yes
TD with finite $\mathcal{S}$	yes/no	NO	NO	YES	YES	NO	NO	YES	YES
TD linear func.approx.	yes/no	NO	NO	yes	yes/no	YES	NO	YES	YES
TD general func.approx.	no/yes	NO	NO	no/yes	NO	YES	NO	YES	YES
Direct Policy Search	no/yes	YES	NO	no/yes	NO	YES	no	YES	YES
Logic Planners	yes/no	YES	yes	YES	YES	no	no	YES	yes
RL with Split Trees	yes	YES	no	YES	NO	yes	YES	YES	YES
Pred.w. Expert Advice	yes/no	YES	–	YES	yes/no	yes	NO	YES	NO
Adaptive Levin Search	no/yes	no	no	yes	yes/no	yes	YES	YES	YES
OOPS	yes/no	no	–	yes	yes/no	YES	YES	YES	YES
Market/Economy RL	yes/no	no	NO	no	yes/no	yes	yes/no	YES	YES
SPXI	no	YES	–	YES	YES	YES	NO	YES	NO
AIXI	NO	YES	YES	yes	YES	YES	YES	YES	YES
AIXItl	no/yes	YES	YES	YES	yes	YES	YES	YES	YES
Human	yes	yes	yes	no/yes	NO	YES	YES	YES	YES

- AIXI can incorporate prior knowledge D: just prepend it in any encoding, it will decrease $K(\mu)$ into $K(\mu|D)$.

- Since AIXI is incomputable but assumes computable environments, it cannot gamble with other AIXIs. Are there interesting environmental classes for which $\mathrm{AI}\xi_{\mathcal{M}} \in \mathcal{M}$ or $\mathrm{AI}\xi\mathrm{tl}_{\mathcal{M}} \in \mathcal{M}$?

Speed Prior and OOPS ⟨reference|TheNewAI⟩

Speed Prior

Assumption. The environment is deterministic.

Postulate. The cumulative prior probability measure of all x incomputable within time t by any method is at most inversely proportional to t.

Algorithm. Set $t := 1$. Start a universal TM with empty input tape. Repeat: While the number of instructions executed so far exceeds t:

$$\text{unbiased coin toss:} \begin{cases} \text{heads up: set } t := 2\,t \\ \text{otherwise exit} \end{cases}$$

If the input contains a symbol, execute it, otherwise set set cell's symbol randomly and set $t := t/2$.

Optimal Ordered Problem Solver

A searcher is n-bias-optimal ($n \geqslant 1$) if for any maximal total search time $T_{\max} > 0$ it is guaranteed to solve any problem $r \in \mathcal{R}$ if the problem has a solution $p \in \mathcal{C}$ that can be created and tested in time $t(p, r) \leqslant P(p|r)\ T_{\max}/n$ (P is the task-specific bias). Basic ingredients of OOPS:

Primitives. Interruptible low- or high-level instructions-tokens (e.g. theorem provers, matrix operators for neural nets, etc.).

Task-specific prefix codes. Token sequences / program prefixes in domain-specific language. Instructions can transfer control to previously selected tokens (loops / calls). A prefix is elongated on program's explicit request. A prefix may be a complete program for some task (programs are prefix-free w.r.t. a task), but may request more tokens on another task (incrementally growing self-delimiting programs).

Access to previous solutions. Let p^n denote a found prefix solving the first n tasks. p^1, ..., p^n are stored or frozen in nonmodifiable memory shared by all tasks (accessible to p^{n+1}) (but can be copied into modifiable task-specific memory).

Bias. Initial task-dependent, user-provided probability distribution on program prefixes.

Self-computed suffix probabilities. Any executed prefix can assign probability distribution to its continuations. Distribution is encoded and manipulated in task-specific internal memory.

Two searches. Run in parallel until p^{n+1} is discovered. The first is exhaustive: tests all possible prefixes in parallel on all tasks up to $n + 1$. The second is focused: searches for prefixes starting with p^n and tests only on task $n+1$ (such prefixes already solve tasks up to n). When optimal solver is found as p^{n_0}, at most half of future run time is wasted – by the first search.

Bias-optimal backtracing. Depth-first search in program space, with backtracing triggered by running over time (prefix probability multiplied by total search time so far). Space is reused.

Example / experiments. Interpreter for FORTH-like language with recursive functions, loops, arithmetic, bias-shifting instructions, domain-specific instructions.

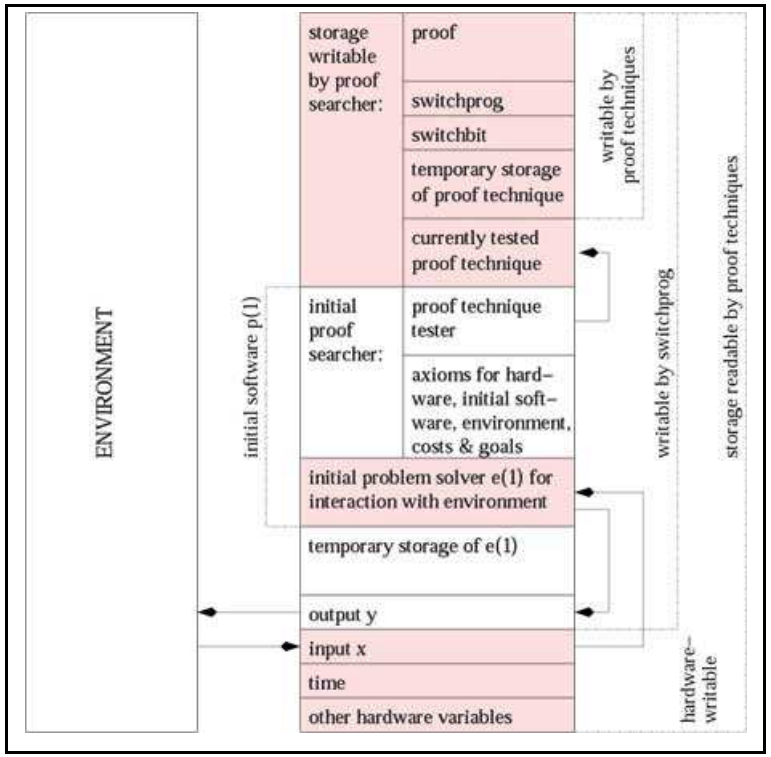
- First taught about recursion: samples of CF language $\{1^k 2^k\}$, $k \leqslant 30$. This took 1/3 of a day. (OOPS found universal solver for all k.)
- Then, by rewriting its search procedure, it learned k-disk "Towers of Hanoi" within a couple of days.

OOPS-Based Reinforcement Learning. Two OOPS modules:

1. The predictor is first trained to find a better world model.
2. The second module (control program) will then use the model to search for a future action sequence with better cumulative reward.
3. After current cycle's time for control program is finished, we will execute the current action of the best control program found in step 2.

OOPS is 8-bias-optimal.

Goedel Machine ⟨reference|GoedelMachines⟩



While executing some initial problem solving strategy, Goedel Machine simultaneously runs a proof searcher which systematically and repeatedly tests proof techniques. An unguarded part of GM `switchprog` can rewrite the whole GM. It is only executed when the GM has found a proof, that it will result in bigger expected reward.

Globally Optimal Self-Changes. Given any formalizable utility function u and assuming consistency of the underlying formal system $\mathcal{A}$, any self-change of p obtained through execution of some `switchprog` identified through the proof of a target theorem [that running `switchprog` increases expected reward] is globally optimal: the utility of executing the present `switchprog` is higher than the utility of waiting for the proof searcher to produce an alternative `switchprog` later.

A program that prints itself.

There is no problem with part of a program to represent the whole program, to any "degree of accuracy".

```
main(){char q=34, n=10,*a="main() {char q=34,n=10,*a=%c%s%c;
printf(a,q,a,q,n);}%c";printf(a,q,a,q,n);}
```

Proof Techniques

1. **get-axiom(n)**
    a. **Hardware axioms**
    b. **Reward axioms**
    c. **Environment axioms**
    d. **Uncertainty axioms** and **string manipulation axioms**
    e. **Initial state axioms**
    f. **Utility axioms**
2. **apply-rule(k, m, n)**
3. **delete-theorem(m)**
4. **set-switchprog(m,n)**
5. **state2theorem(m, n)**

The GM hardware can itself be probabilistic, this has to be represented by a probabilistic logic and in exptectations about which theorems are.

Possible Types of Goedel Machine Self-improvements

1. Just change the ratio of time-sharing between the proof searching subroutine and the subpolicy e–those parts of p responsible for environment interaction.
2. Modify e only. For example, to conduct some experiments and use the resulting knowledge. (Even if it turns out that it would have been better to stick with previous routine, the expectation of reward can favor experimentation.)
3. Modify the axioms to speed up theorem proving.
4. Modify the utility function and target theorem, so that the new values are better according to current target theorem.
5. Modify the probability distribution on proof techniques, etc.
6. Do promptly a very limited rewrite to meet some deadline.
7. In certain uninteresting environments, trash almost all of the GM and leave a looping call to a pleasure center-activating function.
8. Take actions in external environment to augment the machine's hardware.