Universal AI: Solomonoff Induction

David Quarel

Iliad Intensive

June 2026

The big picture: one agent, two halves

- **Universal AI** asks: what would an *optimally intelligent* agent look like, given unlimited compute and the weakest possible assumptions?

The big picture: one agent, two halves

- **Universal AI** asks: what would an *optimally intelligent* agent look like, given unlimited compute and the weakest possible assumptions?
- It splits into two problems:
  1. **Prediction (Solomonoff):** passively observe a sequence $x_1, x_2, \ldots$ and predict the next symbol. *How should you learn?*

The big picture: one agent, two halves

- **Universal AI** asks: what would an *optimally intelligent* agent look like, given unlimited compute and the weakest possible assumptions?
- It splits into two problems:
  1. **Prediction (Solomonoff):** passively observe a sequence $x_1, x_2, \ldots$ and predict the next symbol. *How should you learn?*
  2. **Action (AIXI):** interact with an unknown environment, take actions, collect reward. *How should you act?*

The big picture: one agent, two halves

- **Universal AI** asks: what would an *optimally intelligent* agent look like, given unlimited compute and the weakest possible assumptions?
- It splits into two problems:
  1. **Prediction (Solomonoff):** passively observe a sequence $x_1, x_2, \ldots$ and predict the next symbol. *How should you learn?*
  2. **Action (AIXI):** interact with an unknown environment, take actions, collect reward. *How should you act?*
- **The common engine:** a single **Bayesian mixture** ξ over a class $\mathcal{M}$ of candidate environments, weighted by a prior w_ν.
  - Solomonoff: ξ (eventually) predicts as well as the truth μ.
  - AIXI: the Bayes-optimal policy π^*_ξ (eventually) acts as well as the optimal policy π^*_μ for true environment μ.

The big picture: one agent, two halves

- **Universal AI** asks: what would an *optimally intelligent* agent look like, given unlimited compute and the weakest possible assumptions?
- It splits into two problems:
  1. **Prediction (Solomonoff):** passively observe a sequence $x_1, x_2, \ldots$ and predict the next symbol. *How should you learn?*
  2. **Action (AIXI):** interact with an unknown environment, take actions, collect reward. *How should you act?*
- **The common engine:** a single **Bayesian mixture** ξ over a class $\mathcal{M}$ of candidate environments, weighted by a prior w_ν.
  - Solomonoff: ξ (eventually) predicts as well as the truth μ.
  - AIXI: the Bayes-optimal policy π^*_ξ (eventually) acts as well as the optimal policy π^*_μ for true environment μ.
- **Universal** choice: $\mathcal{M} =$ all computable environments, prior $w_\nu = 2^{-K(\nu)}$ (K is the "complexity" of the environment). Assumption "$\mu \in \mathcal{M}$" becomes "*the universe is computable*".

The problem of induction, formalized

- A true (unknown) environment μ generates a binary sequence; each bit $x_t \sim \mu(\cdot \mid x_{<t})$.

The problem of induction, formalized

- A true (unknown) environment μ generates a binary sequence; each bit $x_t \sim \mu(\cdot \mid x_{<t})$.
- A **predictor** P sees the past $x_{<t}$ and outputs a distribution $P(\cdot \mid x_{<t})$ over the next bit.

The problem of induction, formalized

- A true (unknown) environment μ generates a binary sequence; each bit $x_t \sim \mu(\cdot \mid x_{<t})$.
- A **predictor** P sees the past $x_{<t}$ and outputs a distribution $P(\cdot \mid x_{<t})$ over the next bit.
- **Quality measure:** Cumulative μ-expected squared prediction error $S_\infty^\mu := \sum_{t=1}^{\infty} s_t^\mu$

Per-step squared prediction error s_t^μ

$$s_t^\mu = \mathbb{E}_\mu \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right] = \sum_{x_{<t}} \mu(x_{<t}) \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right]$$

The problem of induction, formalized

- A true (unknown) environment μ generates a binary sequence; each bit $x_t \sim \mu(\cdot \mid x_{<t})$.
- A **predictor** P sees the past $x_{<t}$ and outputs a distribution $P(\cdot \mid x_{<t})$ over the next bit.
- **Quality measure:** Cumulative μ-expected squared prediction error $S_\infty^\mu := \sum_{t=1}^{\infty} s_t^\mu$

Per-step squared prediction error s_t^μ

$$s_t^\mu = \mathbb{E}_\mu \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right] = \sum_{x_{<t}} \mu(x_{<t}) \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right]$$

- **Goal:** build a P that makes cumulative error S_∞^μ *small*.

The problem of induction, formalized

- A true (unknown) environment μ generates a binary sequence; each bit $x_t \sim \mu(\cdot \mid x_{<t})$.
- A **predictor** P sees the past $x_{<t}$ and outputs a distribution $P(\cdot \mid x_{<t})$ over the next bit.
- **Quality measure:** Cumulative μ-expected squared prediction error $S_\infty^\mu := \sum_{t=1}^{\infty} s_t^\mu$

Per-step squared prediction error s_t^μ

$$s_t^\mu = \mathbb{E}_\mu \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right] = \sum_{x_{<t}} \mu(x_{<t}) \left[\sum_{x_t} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \right]$$

- **Goal:** build a P that makes cumulative error S_∞^μ *small*.
- **Idea:** don't bet on one model. Hedge your bets and mix over a whole class $\mathcal{M} = \{\nu_1, \nu_2, \dots\}$ of potential environments.

The Bayesian mixture ξ

Definition

Class $\mathcal{M} = \{\nu_1, \nu_2, \dots\}$ with prior weights $w_\nu > 0$, $\sum_\nu w_\nu = 1$.

$$\xi(x_{1:t}) := \sum_{\nu \in \mathcal{M}} w_\nu \, \nu(x_{1:t}).$$

The Bayesian mixture ξ

Definition

Class $\mathcal{M} = \{\nu_1, \nu_2, \dots\}$ with prior weights $w_\nu > 0$, $\sum_\nu w_\nu = 1$.

$$\xi(x_{1:t}) := \sum_{\nu \in \mathcal{M}} w_\nu \, \nu(x_{1:t}).$$

- **Key fact (mixture is a predictor):** its one-step prediction is a *posterior-weighted* average of the members' predictions (you prove this today!)

$$\xi(x_t \mid x_{<t}) = \sum_{\nu \in \mathcal{M}} w(\nu \mid x_{<t}) \, \nu(x_t \mid x_{<t}), \qquad w(\nu \mid x_{<t}) = w_\nu \, \frac{\nu(x_{<t})}{\xi(x_{<t})}.$$

The Bayesian mixture ξ

Definition

Class $\mathcal{M} = \{\nu_1, \nu_2, \dots\}$ with prior weights $w_\nu > 0$, $\sum_\nu w_\nu = 1$.

$$\xi(x_{1:t}) := \sum_{\nu \in \mathcal{M}} w_\nu \, \nu(x_{1:t}).$$

- **Key fact (mixture is a predictor):** its one-step prediction is a *posterior-weighted* average of the members' predictions (you prove this today!)

$$\xi(x_t \mid x_{<t}) = \sum_{\nu \in \mathcal{M}} w(\nu \mid x_{<t}) \, \nu(x_t \mid x_{<t}), \qquad w(\nu \mid x_{<t}) = w_\nu \, \frac{\nu(x_{<t})}{\xi(x_{<t})}.$$

- **Posterior update is multiplicative:** $w(\nu \mid x_{1:t}) = w(\nu \mid x_{<t}) \cdot \dfrac{\nu(x_t \mid x_{<t})}{\xi(x_t \mid x_{<t})}$: we reweight by how well ν predicted vs. ξ (Bayes' rule!)

Main result: the cumulative prediction bound

Cumulative bound

$$S_\infty^\mu := \sum_{t=1}^{\infty} \sum_{x_{<t}} \mu(x_{<t}) \sum_{x_t \in \mathbb{B}} \left(P(x_t \mid x_{<t}) - \mu(x_t \mid x_{<t}) \right)^2 \leq -\ln w_\mu$$

- Total squared error over *all time* is finite, bounded by the prior penalty on μ.
- Higher prior on the truth $\Rightarrow$ smaller total error.

Specializing to the Solomonoff prior

- Take the **Solomonoff prior** $w_\nu = 2^{-K(\nu)}$, where $K(\nu)$ is the *Kolmogorov complexity*: length of the shortest program computing ν.

Specializing to the Solomonoff prior

- Take the **Solomonoff prior** $w_\nu = 2^{-K(\nu)}$, where $K(\nu)$ is the *Kolmogorov complexity*: length of the shortest program computing ν.
- Substitute $w_\mu = 2^{-K(\mu)}$ into the cumulative bound:

Explicit complexity bound

$$S_\infty^\mu \;\leq\; K(\mu)\ln 2$$

Specializing to the Solomonoff prior

- Take the **Solomonoff prior** $w_\nu = 2^{-K(\nu)}$, where $K(\nu)$ is the *Kolmogorov complexity*: length of the shortest program computing ν.
- Substitute $w_\mu = 2^{-K(\mu)}$ into the cumulative bound:

Explicit complexity bound

$$S_\infty^\mu \;\leq\; K(\mu)\ln 2$$

- **Interpretation:** the total prediction error is bounded by the *description length* of the true environment. Simpler worlds are learned faster.
- This is the formal version of **Occam's razor**.

Other properties of the Solomonoff prior

- **Bounded mistakes:** on a deterministic sequence, predicting the most likely bit $\hat{x}_t = \arg\max_{x \in \mathbb{B}} \xi(x \mid x_{<t})$ makes at most $-2 \ln w_\mu$ wrong guesses *ever*.

Other properties of the Solomonoff prior

- **Bounded mistakes:** on a deterministic sequence, predicting the most likely bit $\hat{x}_t = \arg\max_{x \in \mathbb{B}} \xi(x \mid x_{<t})$ makes at most $-2 \ln w_\mu$ wrong guesses *ever*.
- **Pareto optimality:** no other predictor beats ξ on *every* $\nu \in \mathcal{M}$ simultaneously (for KL loss *and* squared loss).

Other properties of the Solomonoff prior

- **Bounded mistakes:** on a deterministic sequence, predicting the most likely bit $\hat{x}_t = \arg\max_{x \in \mathbb{B}} \xi(x \mid x_{<t})$ makes at most $-2 \ln w_\mu$ wrong guesses *ever*.
- **Pareto optimality:** no other predictor beats ξ on *every* $\nu \in \mathcal{M}$ simultaneously (for KL loss *and* squared loss).
- **Misspecification ($\mu \notin \mathcal{M}$):** the bound degrades gracefully,

$$S_\infty^\mu \leq -\ln w_{\hat{\mu}} + D_n(\mu \parallel \hat{\mu})$$

where $\hat{\mu}$ is the closest model in $\mathcal{M}$ to μ.

Takeaways

- **Solomonoff** solves induction: mix over all computable hypotheses, weighted by simplicity. Total prediction error $\leq K(\mu) \ln 2$.

Takeaways

- **Solomonoff** solves induction: mix over all computable hypotheses, weighted by simplicity. Total prediction error $\leq K(\mu) \ln 2$.
- **AIXI** lifts this to action: the Bayes-optimal policy over the same universal mixture. It learns to predict *and* to act as well as if it knew the world.

Takeaways

- **Solomonoff** solves induction: mix over all computable hypotheses, weighted by simplicity. Total prediction error $\leq K(\mu) \ln 2$.
- **AIXI** lifts this to action: the Bayes-optimal policy over the same universal mixture. It learns to predict *and* to act as well as if it knew the world.
- **The catch:** both are *incomputable* (K is incomputable; $\mathcal{M}$ is intractable to enumerate). They are gold standards, not computable algorithms, and the prior's $2^{-K(\nu)}$ hides a dependency on the choice of universal Turing machine.

Takeaways

- **Solomonoff** solves induction: mix over all computable hypotheses, weighted by simplicity. Total prediction error $\leq K(\mu) \ln 2$.
- **AIXI** lifts this to action: the Bayes-optimal policy over the same universal mixture. It learns to predict *and* to act as well as if it knew the world.
- **The catch:** both are *incomputable* (K is incomputable; $\mathcal{M}$ is intractable to enumerate). They are gold standards, not computable algorithms, and the prior's $2^{-K(\nu)}$ hides a dependency on the choice of universal Turing machine.
- **Why care for safety:** a precise, assumption-light model of idealized intelligence providing a lens on what optimal agents do. For example, we can show under what circumstances AIXI would wirehead [RO11], safely rather than recklessly explore [CCH20], or self-modify [OR11] as a model of what superintelligent agents would do.

References I

[CCH20] Michael K. Cohen, Elliot Catt, and Marcus Hutter.
Curiosity killed or incapacitated the cat and the asymptotically optimal agent.
CoRR, abs/2006.03357, 2020.
URL: https://arxiv.org/abs/2006.03357.

[OR11] Laurent Orseau and Mark Ring.
Self-modification and mortality in artificial agents.
In *Artificial General Intelligence (AGI 2011)*, volume 6830 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2011.
URL: https://link.springer.com/chapter/10.1007/978-3-642-22887-2_1, doi:10.1007/978-3-642-22887-2_1.

[RO11] Mark Ring and Laurent Orseau.
Delusion, survival, and intelligent agents.
In *Artificial General Intelligence (AGI 2011)*, volume 6830 of *Lecture Notes in Computer Science*, pages 11–20. Springer, 2011.
URL: https://people.idsia.ch/~ring/AGI-2011/Paper-B.pdf, doi:10.1007/978-3-642-22887-2_2.