Reflective variants of Solomonoff induction and AIXI

Benja Fallenstein, Nate Soares and Jessica Taylor

Machine Intelligence Research Institute

July 24, 2015

Motivation

- What does it mean to **learn optimally in the real world**?
    - Closest thing to a definition:
    - Solomonoff induction and AIXI

Motivation

- What does it mean to **learn optimally in the real world**?
  - Closest thing to a definition:
  - Solomonoff induction and AIXI
- Environments can't contain other equally powerful systems
  - But that's important in the real world!

Motivation

- What does it mean to **learn optimally in the real world**?
  - Closest thing to a definition:
  - Solomonoff induction and AIXI
- Environments can't contain other equally powerful systems
  - But that's important in the real world!
- Seems like a pretty fundamental flaw
  - In order to figure out what the environment does, need more computing power than this environment

Motivation

- What does it mean to **learn optimally in the real world**?
  - Closest thing to a definition:
  - Solomonoff induction and AIXI
- Environments can't contain other equally powerful systems
  - But that's important in the real world!
- Seems like a pretty fundamental flaw
  - In order to figure out what the environment does, need more computing power than this environment
  - Halting oracles can't talk about machines with halting oracles

Motivation

- What does it mean to **learn optimally in the real world**?
  - Closest thing to a definition:
  - Solomonoff induction and AIXI
- Environments can't contain other equally powerful systems
  - But that's important in the real world!
- Seems like a pretty fundamental flaw
  - In order to figure out what the environment does, need more computing power than this environment
  - Halting oracles can't talk about machines with halting oracles
- But actually. . .

Why SI can't predict itself

- Solomonoff induction (SI), roughly:
  - Predict infinite bitstrings.
  - Hypotheses: any program which outputs an infinite bitstring.
  - Prior probability $\propto 2^{-\text{length of program}}$

Why SI can't predict itself

- Solomonoff induction (SI), roughly:
  - Predict infinite bitstrings.
  - Hypotheses: any program which outputs an infinite bitstring.
  - Prior probability $\propto 2^{-\text{length of program}}$
- For each hypothesis, SI must compute next bit.
  - SI mustn't loop, even if a hypothesis loops.
  - Needs halting oracle.
  - But halting oracle only takes machines *without* a halting oracle.

Why SI can't predict itself

- Solomonoff induction (SI), roughly:
  - Predict infinite bitstrings.
  - Hypotheses: any program which outputs an infinite bitstring.
  - Prior probability $\propto 2^{-\text{length of program}}$
- For each hypothesis, SI must compute next bit.
  - SI mustn't loop, even if a hypothesis loops.
  - Needs halting oracle.
  - But halting oracle only takes machines *without* a halting oracle.
- Attempt to fix:
  - Oracle that returns 0/1 if hypothesis returns 0/1
  - Can return either 0 or 1 if program loops
  - But: Ask "what do I return" and return opposite

Reflective oracles

- *Probabilistic oracle machines*:
  - Turing machines which can (1) flip coins and (2) call an oracle.
  - The oracle may answer randomly.
  - $\mathbb{P}[M^O() = 1] =$ prob. that M returns 1 when run on oracle O.

Reflective oracles

- *Probabilistic oracle machines*:
  - Turing machines which can (1) flip coins and (2) call an oracle.
  - The oracle may answer randomly.
  - $\mathbb{P}[M^O() = 1]$ = prob. that M returns 1 when run on oracle O.
- Reflective oracles:
  - $O(M, x, p)$: M machine, x input, p probability
  - Always returns 0 or 1, possibly probabilistically.
  - $\mathbb{P}[M^O(x) = 1] > p \implies \mathbb{P}[O(M, x, p) = 1] = 1$
  - $\mathbb{P}[M^O(x) = 0] > 1 - p \implies \mathbb{P}[O(M, x, p) = 0] = 1$

Reflective oracles

- *Probabilistic oracle machines*:
  - Turing machines which can (1) flip coins and (2) call an oracle.
  - The oracle may answer randomly.
  - $\mathbb{P}[M^O() = 1]$ = prob. that M returns 1 when run on oracle O.
- Reflective oracles:
  - $O(M, x, p)$: M machine, x input, p probability
  - Always returns 0 or 1, possibly probabilistically.
  - $\mathbb{P}[M^O(x) = 1] > p \implies \mathbb{P}[O(M, x, p) = 1] = 1$
  - $\mathbb{P}[M^O(x) = 0] > 1 - p \implies \mathbb{P}[O(M, x, p) = 0] = 1$
- E.g.: Ask oracle what I do and do the opposite
  - $M^O() := 1 - O(M, \epsilon, 0.5)$

Reflective oracles

- *Probabilistic oracle machines*:
  - Turing machines which can (1) flip coins and (2) call an oracle.
  - The oracle may answer randomly.
  - $\mathbb{P}[M^O() = 1]$ = prob. that M returns 1 when run on oracle O.
- Reflective oracles:
  - $O(M, x, p)$: M machine, x input, p probability
  - Always returns 0 or 1, possibly probabilistically.
  - $\mathbb{P}[M^O(x) = 1] > p \implies \mathbb{P}[O(M, x, p) = 1] = 1$
  - $\mathbb{P}[M^O(x) = 0] > 1 - p \implies \mathbb{P}[O(M, x, p) = 0] = 1$
- E.g.: Ask oracle what I do and do the opposite
  - $M^O() := 1 - O(M, \epsilon, 0.5)$
  - Not a contradiction: $\mathbb{P}[M^O() = 1] = 0.5$

Reflective Solomonoff induction

- Hypothesis = machine, takes bitstring so far, returns next bit
  - $\implies$ $O(M, x, p)$ talks about *conditional probability* given x

Reflective Solomonoff induction

- Hypothesis = machine, takes bitstring so far, returns next bit
  - $\implies O(M, x, p)$ talks about *conditional probability* given x
- Given machine $M^O(x)$ that may loop, construct $N^O(x)$:
  - Flip a fair coin. If heads: Return $O(M, x, 0.5)$.
  - If tails: Run $O(M, x, 0.5)$; depending on result, replace 0.5 by either 0.25 or 0.75; start from beginning (binary search)
  - $N^O(x)$ never loops, and: $\mathbb{P}[N^O(x) = b] \geq \mathbb{P}[M^O(x) = b]$

Reflective Solomonoff induction

- Hypothesis = machine, takes bitstring so far, returns next bit
  - $\implies O(M, x, p)$ talks about *conditional probability* given x
- Given machine $M^O(x)$ that may loop, construct $N^O(x)$:
  - Flip a fair coin. If heads: Return $O(M, x, 0.5)$.
  - If tails: Run $O(M, x, 0.5)$; depending on result, replace 0.5 by either 0.25 or 0.75; start from beginning (binary search)
  - $N^O(x)$ never loops, and: $\mathbb{P}[N^O(x) = b] \geq \mathbb{P}[M^O(x) = b]$
- Reflective Solomonoff induction $\mathrm{rSI}^O(x)$:
  - Sample non-looping machine N^O
  - Rejection sampling: compute probability that N^O produces string x, accept N^O with this probability, else start over
  - Output $N^O(x)$

Reflective Solomonoff induction and AIXI

- $\mathrm{rSI}^O(x)$ can reason about worlds making calls to $\mathrm{rSI}^O(x)$
  - E.g.: environment that outputs bit $\mathrm{rSI}^O(x)$ considers less likely
  - **(You could do this to a real-world predictor!)**

Reflective Solomonoff induction and AIXI

- $\mathrm{rSI}^O(x)$ can reason about worlds making calls to $\mathrm{rSI}^O(x)$
  - E.g.: environment that outputs bit $\mathrm{rSI}^O(x)$ considers less likely
  - **(You could do this to a real-world predictor!)**
  - Thm: $\mathrm{rSI}^O(x)$ converges to perfect predictions on any oracle-computable environment (including this one)
  - Solution: probability that next bit is 1 is 0.5

Reflective Solomonoff induction and AIXI

- $\mathrm{rSI}^O(x)$ can reason about worlds making calls to $\mathrm{rSI}^O(x)$
  - E.g.: environment that outputs bit $\mathrm{rSI}^O(x)$ considers less likely
  - **(You could do this to a real-world predictor!)**
  - Thm: $\mathrm{rSI}^O(x)$ converges to perfect predictions on any oracle-computable environment (including this one)
  - Solution: probability that next bit is 1 is 0.5
- Analogously, can define reflective variant of AIXI
  - Has hypotheses containing other reflective AIXIs
  - If it learns that it is in one of these worlds:
    - (roughly) Plays a Nash equilibrium.

Reflective Solomonoff induction and AIXI

- $\mathrm{rSI}^O(x)$ can reason about worlds making calls to $\mathrm{rSI}^O(x)$
  - E.g.: environment that outputs bit $\mathrm{rSI}^O(x)$ considers less likely
  - **(You could do this to a real-world predictor!)**
  - Thm: $\mathrm{rSI}^O(x)$ converges to perfect predictions on any oracle-computable environment (including this one)
  - Solution: probability that next bit is 1 is 0.5
- Analogously, can define reflective variant of AIXI
  - Has hypotheses containing other reflective AIXIs
  - If it learns that it is in one of these worlds:
    - (roughly) Plays a Nash equilibrium.
- Reflective oracle existence proof is closely related to Nash eq.

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)
  - Real-world: environment can contain equally powerful systems
    - Reflective AIXI and SI extend definiton to deal with this

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)
  - Real-world: environment can contain equally powerful systems
    - Reflective AIXI and SI extend definiton to deal with this
- But the real world doesn't contain reflective oracles!

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)
  - Real-world: environment can contain equally powerful systems
    - Reflective AIXI and SI extend definiton to deal with this
- But the real world doesn't contain reflective oracles!
  - Real-world systems will need to have *uncertainty about mathematical facts* in predictions about other predictors

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)
  - Real-world: environment can contain equally powerful systems
    - Reflective AIXI and SI extend definiton to deal with this
- But the real world doesn't contain reflective oracles!
  - Real-world systems will need to have *uncertainty about mathematical facts* in predictions about other predictors
  - Reflective SI/AIXI as *a step towards* theory of very intelligent systems reasoning about very intelligent systems

Conclusions

- AIXI and SI are definitions of *perfect* agents and predictors.
  - (IMO not exactly right, but a large step forward.)
  - Real-world: environment can contain equally powerful systems
    - Reflective AIXI and SI extend definiton to deal with this
- But the real world doesn't contain reflective oracles!
  - Real-world systems will need to have *uncertainty about mathematical facts* in predictions about other predictors
  - Reflective SI/AIXI as *a step towards* theory of very intelligent systems reasoning about very intelligent systems
- Thank you for your attention!