An attempt to formalize counterpossible reasoning in deterministic settings

Nate Soares and Benja Fallenstein

Machine Intelligence Research Institute

July 23, 2015

What is a "what if"?

- To make decisions,
    - need not only a good model of the *real* world
    - but also of what *would* happen if you *did something else*.

What is a "what if"?

- To make decisions,
  - need not only a good model of the *real* world
  - but also of what *would* happen if you *did something else*.
- From Solomonoff induction (SI) to AIXI:
  - SI uses **simplest explanation** of **what actually happened** in the past to predict **will actually happen** in the future.
    - Guaranteed to converge if environment is computable.

What is a "what if"?

- To make decisions,
  - need not only a good model of the *real* world
  - but also of what *would* happen if you *did something else*.
- From Solomonoff induction (SI) to AIXI:
  - SI uses **simplest explanation** of **what actually happened** in the past to predict **will actually happen** in the future.
    - Guaranteed to converge if environment is computable.
  - AIXI uses **simplest explanation** of **what actually happened** in the past to predict **what would happen** if the agent took different actions.

What is a "what if"?

- To make decisions,
  - need not only a good model of the *real* world
  - but also of what *would* happen if you *did something else*.
- From Solomonoff induction (SI) to AIXI:
  - SI uses **simplest explanation** of **what actually happened** in the past to predict **will actually happen** in the future.
    - Guaranteed to converge if environment is computable.
  - AIXI uses **simplest explanation** of **what actually happened** in the past to predict **what would happen** if the agent took different actions.
    - Not guaranteed to converge.

What is a "what if"?

- To make decisions,
  - need not only a good model of the *real* world
  - but also of what *would* happen if you *did something else*.
- From Solomonoff induction (SI) to AIXI:
  - SI uses **simplest explanation** of **what actually happened** in the past to predict **will actually happen** in the future.
    - Guaranteed to converge if environment is computable.
  - AIXI uses **simplest explanation** of **what actually happened** in the past to predict **what would happen** if the agent took different actions.
    - Not guaranteed to converge.
    - Is this the definition of "what if" we want?

Agents are part of the world

- AIXI sees the agent as separate from the environment.
  - "What if" is defined by well-defined I/O channels.
- In reality, the agent is *part* of its environment.

Agents are part of the world

- AIXI sees the agent as separate from the environment.
  - "What if" is defined by well-defined I/O channels.
- In reality, the agent is *part* of its environment.
  - An agent's actions can disrupt its computations.
  - Its internals can affect the environment (e.g., heating up).

Agents are part of the world

- AIXI sees the agent as separate from the environment.
  - "What if" is defined by well-defined I/O channels.
- In reality, the agent is *part* of its environment.
  - An agent's actions can disrupt its computations.
  - Its internals can affect the environment (e.g., heating up).
  - Which transistors in the CPU change in the counterfactual world where the agent outputs a different action?

Agents are part of the world

- AIXI sees the agent as separate from the environment.
  - "What if" is defined by well-defined I/O channels.
- In reality, the agent is *part* of its environment.
  - An agent's actions can disrupt its computations.
  - Its internals can affect the environment (e.g., heating up).
  - Which transistors in the CPU change in the counterfactual world where the agent outputs a different action?
- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
  - Is the thing AIXI would learn the definition we want?

Prisoner's Dilemma with shared source code

- Two agents can each choose to pay \$1 ("cooperate").
- If an agent cooperates, the other agent receives \$2.

Prisoner's Dilemma with shared source code

- Two agents can each choose to pay \$1 ("cooperate").
- If an agent cooperates, the other agent receives \$2.
- The agents know each other's source code, and know *that they make decisions in exactly the same* (symmetric) *way*
  - Thus, each cooperates *if and only if* the other does.

Prisoner’s Dilemma with shared source code

- Two agents can each choose to pay $1 (“cooperate”).
- If an agent cooperates, the other agent receives $2.

- The agents know each other’s source code, and know *that they make decisions in exactly the same* (symmetric) *way*
  - Thus, each cooperates *if and only if* the other does.

- Classical game theory:
  - If opponent cooperates, it’s better to defect ($2 vs. $1)
  - If opponent defects, it’s better to defect ($0 vs. −$1)
  - Thus, it’s best to defect

Prisoner's Dilemma with shared source code

- Two agents can each choose to pay \$1 ("cooperate").
- If an agent cooperates, the other agent receives \$2.
- The agents know each other's source code, and know *that they make decisions in exactly the same* (symmetric) *way*
  - Thus, each cooperates *if and only if* the other does.
- Classical game theory:
  - If opponent cooperates, it's better to defect (\$2 vs. \$1)
  - If opponent defects, it's better to defect (\$0 vs. −\$1)
  - Thus, it's best to defect
- Intuition: If I cooperate, so does opponent (\$1 vs. \$0)!

Abstract reasoning

- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
- In Prisoner's Dilemma, we use *abstract reasoning*.
  - We don't run the opponent's computation.
  - Instead, use abstract argument to show: *If I cooperate, they cooperate. If I defect, they defect.*

Abstract reasoning

- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
- In Prisoner's Dilemma, we use *abstract reasoning*.
  - We don't run the opponent's computation.
  - Instead, use abstract argument to show: *If I cooperate, they cooperate. If I defect, they defect.*
- Realistic generally intelligent agents will use abstract reasoning all the time.

Abstract reasoning

- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
- In Prisoner's Dilemma, we use *abstract reasoning*.
  - We don't run the opponent's computation.
  - Instead, use abstract argument to show: *If I cooperate, they cooperate. If I defect, they defect.*
- Realistic generally intelligent agents will use abstract reasoning all the time.
- Best **mathematical model** of abstract reasoning available today: Formal logic.

Proof-based UDT

- Agents $A_0()$, $A_1()$ with utility functions $U_0()$, $U_1()$.
- def $U_i()$:
  cost := if $A_i() = C$ then 1 else 0
  benefit := if $A_{1-i}() = C$ then 2 else 0
  return benefit $-$ cost
- def $A_i()$:
  For each outcome o, from best to worst $(2, 1, 0, -1)$:
  For each action a in $\{C, D\}$:
  If Peano Arithmetic proves
  $\ulcorner A_i() = a \rightarrow U_i() = o \urcorner$,
  then return a.
  Return default action (e.g., D).

Reasoning under false premises

- Worrying: If $A_1() = D$, then $A_1() = C \to U_1() = -1$.
  - Given a false premise, every conclusion is true!

Reasoning under false premises

- Worrying: If $A_1() = D$, then $A_1() = C \to U_1() = -1$.
  - Given a false premise, every conclusion is true!
- However, this implication is not *provable*.
  - Can prove $A_1() = C \to U_1() = 1$.
  - If can also prove $A_1() = C \to U_1() = -1$,
    - then can prove $A_1() \neq C$
    - and then can prove $A_1() = C \to U_1() = 3$
    - but then $A_1() = C$ (loop terminates immediately)
    - and so $A_1() \neq C$ is false, contradiction.
- For each action a, Peano Arithmetic has *non-standard models* where $A_1() = a$.
  - Can see these as "impossible possible worlds".

A troubling example

- The agent must choose between a \$5 bill and a \$10 bill.
- But there is a twist: *If Peano Arithmetic is inconsistent*, its money will be taken away if it chooses \$10!

- def $U()$:

```
def U():
    if A() = 10:
        if (Peano Arithmetic is consistent):
            return $10
        return $0
    return $5
```

A troubling example

- The agent must choose between a \$5 bill and a \$10 bill.
- But there is a twist: *If Peano Arithmetic is inconsistent*, its money will be taken away if it chooses \$10!

- def $U()$:

```
def U():
    if A() = 10:
        if (Peano Arithmetic is consistent):
            return $10
        return $0
    return $5
```

- Proof-based UDT chooses to take \$5.
  - It turns out that even though Peano Arithmetic is consistent **in the actual world**,
  - it is "inconsistent" in **all impossible possible worlds** in which the agent takes the \$10 bill!

Conclusions

- What is the correct notion of "what if"?
- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
- Proof-based UDT as a model of counterpossible reasoning.
    - Still crazy opinions about the "impossible possible worlds".

Conclusions

- What is the correct notion of "what if"?
- What would happen if **a deterministic subprocess** of a deterministic world **returned a different value** than what it actually returns?
- Proof-based UDT as a model of counterpossible reasoning.
  - Still crazy opinions about the "impossible possible worlds".
- Maybe none of the solutions suggested are correct.
  - But there's still something we'd *want* a smarter-than-human agent to do if we thought about the problem for a very long time.
  - Want to understand what this is, and make sure actual smarter-than-human agents behave reasonably according to this criterion.