Unbiased Prior Belief of Normalized Utility

by Sven Nilsen, 2017

One common mistake I make and also lot of people struggle with, is failing to consider all available information when making a decision. If I could figure out how to avoid this problem just a bit, it might give a huge payoff over long term. In this paper I derive the decision function of evaluating an idea based on an estimate of the number of ideas available and a normalized utility score assigned to the idea to evaluate.

The result of this paper is a critical estimate of normalized utility for any idea among `n` alternatives:

$$u = 2^{-1/n}$$

This simple formula might be used to decide whether any particular idea is worth pursuing, given that there is little or no information about alternative ideas.

Assumptions:

- Ideas are uniformly distributed on a normalized utility scale
- An estimate is available on the number of other ideas
- There is at least one known alternative good idea

An overview of critical utility values with 2 decimals precision for some values of `n`:

N	**U**
1	0.50
2	0.70
3	0.79
4	0.84
5	0.87
6	0.89
7	0.91
8	0.92
9	0.93
10	0.93
100	0.99

As one can observe by taking the limit of `n → ∞`, the chance of making optimal decisions is zero.

The rest of this paper is a proof of the formula.

Assume that a good idea is an object `a` such that:

a : [is_a_good_idea] true

is_a_good_idea : idea → bool

In order for deciding rationally to pursue an idea, it requires at least a 50% chance, as subjective belief, that there is no other idea that gives a higher utility score. Otherwise you pay an opportunity cost for making that decision, as it prevents you from pursuing other ideas in parallel. This assumes a sequential agent architecture.

Translated to path semantics, I want to know the probabilistic path of `is_a_good_idea`, which returns the probability of it returning `true` by constraining the input by `a`:

$$\text{try\_it} := \backslash(a : \text{idea}) = (\exists_p \text{is\_a\_good\_idea}\{(= a)\})(\text{true}) > 1/2$$

Of course, setting the input equal to `a` determines the boolean return value, such that the probability of returning `true` is either 0 or 1. However, this approach is also valid when the idea is constrained by a sub-type, which in path semantics can be any halting function. Making `try_id` a higher order function that takes a knowledge function `f`:

$$\text{try\_it} := \backslash(f : \text{idea} \rightarrow \text{bool}) = \backslash(a : \text{idea}) = (\exists_p \text{is\_a\_good\_idea}\{f(a)\})(\text{true}) > 1/2$$

This has the advantage of controlling ignorance by will, what you pretend to know about an idea, for reflection purposes and self diagnostics. It also generates a sub-problem to solve in order to make optimal decisions given the available information. I choose a function such that `f` returns `true` when there is no other idea that gives a higher utility score:

$$\text{no\_better\_idea} := \backslash(\text{ideas} : [\text{idea}]) = \backslash(\text{this} : \text{idea}) = \neg\exists\ i\ \{\ \text{utility}(\text{ideas}_i) > \text{utility}(\text{this})\ \}$$

decide := \(ideas : [idea]) = try_it(no_better_idea(ideas))
decide : idea → bool

This means constraining the set of ideas to only consider those “good ideas” which utility is expected to be better than other ideas you might have. One might argue that `is_a_good_idea` should be used directly for decision making, since it has the same type `idea → bool`. In practice it is useful to consider a good idea for being a “good idea” in general, and then select those ideas that are appropriate for a given context. This is conceptually simpler and more flexible than trying to figure out whether an idea demands an intrinsic attention to doing it or not, e.g. based on some form of ideology.

Now, when you give the decision function a bad idea:

decide(among_some_ideas)(_ : [is_a_good_idea] false)

The partial function `is_a_good_idea{f(a)}` gets over-constrained once there exists at least some idea with higher utility. Its existential path returns `false` for all inputs. Therefore, the probabilistic path will return `0` on all inputs and since this is lower than 1/2 (50%), the `try_it` function always returns `false`. This is a proof that you should not try bad ideas when you got something else to do.

A problem is that, while making optimal decisions possible for a given context, assigning a utility to every idea requires time and energy. I would like to have a way to estimate whether pursuing some idea is worth it without comparing it to every other idea. One approach to this problem is to use a normalized scale of utility and apply probability theory. For simplicity, I assume a uniform distribution of ideas across the scale between 0 and 1. Deriving the probability is easy:

$$
\begin{aligned}
&P(\neg\exists\ i\ \{\ \mathrm{utility}(\mathrm{ideas}_i) > \mathrm{utility}(\mathrm{this})\ \})\\
&1 - P(\exists\ i\ \{\ \mathrm{utility}(\mathrm{ideas}_i) > \mathrm{utility}(\mathrm{this})\ \})\\
&1 - (1 - \prod\ i\ \{\ 1\text{-}P(\mathrm{utility}(\mathrm{ideas}_i) > \mathrm{utility}(\mathrm{this}))\ \})\\
&1 - (1 - \prod\ i\ \mathrm{len}(\mathrm{ideas})\ \{\ 1\text{-}(1\text{-}\mathrm{utility}(\mathrm{this}))\ \})\\
&1 - (1 - \prod\ i\ \mathrm{len}(\mathrm{ideas})\ \{\ \mathrm{utility}(\mathrm{this})\ \})\\
&1 - (1 - \mathrm{utility}(\mathrm{this})^{\mathrm{len}(\mathrm{ideas})})\\
&\mathrm{utility}(\mathrm{this})^{\mathrm{len}(\mathrm{ideas})}
\end{aligned}
$$

A problem is that constraining the input of `is_a_good_idea` by probability does not work, so I take a look at the overall expression:

$$(\exists_p \mathrm{is\_a\_good\_idea}\{f(a)\})(\mathrm{true}) > 1/2$$

If I assume that I have at least one good idea, and I never assign higher utility to good ideas compared to bad ideas e.g. because my moral code is reflected in the utility function, then the idea with highest utility score should be a good idea. Therefore, `is_a_good_idea` should return `true` and when it does, the probabilistic path returns `1`. The confidence in the particular idea being the idea with highest utility score should be multiplied with `1`, so I end up with:

$$\mathrm{try\_it} := \backslash(\_ : \mathrm{idea} \rightarrow \mathrm{bool} \wedge (= \mathrm{no\_better\_idea}(\_ : [\mathrm{len}]\ n)) = \backslash(\_ : \mathrm{idea} \wedge [\mathrm{utility}]\ u) = u^n > 1/2$$

Finding the critical value for `u`:

$$
\begin{aligned}
&u^n = 1/2\\
&u = 2^{-1/n}
\end{aligned}
$$

So, if you have 10 ideas, any particular idea worth considering should at least have a 93.3% utility.