A Logical System of Emotional Desire

Chris Broski, 2017

Abstract

Intelligent systems can exhibit the desire to accomplish a goal in several ways. *Instinctual desire* is an emergent property of physical form and programmed behavior. *Emotional desire* exists if a creature possesses subjective sensors that indicate whether situations are good or bad, and uses these as a guide to modify its own behavior toward increased benefit and decreased harm. *Conscious desire* can occur if a system has a model of how its actions affect the environment and can simulate the benefit or harm of possible consequences before acting. I describe the logical data structures and processes required to create a simple system of emotional desire.

About Mathematical Notation

Though this essay is about condensing broad concepts of intelligent behavior to simple mathematical models, I have chosen to not use formal mathematical notation in my examples. Formal math is not executable by a computer in a standardized way, nor is it intuitively comprehended by non-mathematicians. (It is also not easily typed on a keyboard.) I have decided to document my ideas using a computer programming language. My chosen language is JavaScript (ES5 standard) for the following reasons:

1. It is a very high-level language, so it does not require machine-specific instructions such as memory management.
2. It is a commonly used language, especially by those whose primary expertise is not computer programming.
3. It can be executed natively on any computer with a current web browser.

Because of the "duck-typing" of numeric values in JavaScript, floating-point data will be indicated by the inclusion of a decimal point. Numbers without a decimal point should be considered integers. To keep numbers small and easy to read, significant figures are not intended to indicate precision.

What is a Goal?

I hate to start any discussion without clearly stating definitions of key concepts.

What does *goal* mean, rigorously and precisely? Like most people, when I need a word defined, I consult a dictionary. My favorite definition of *goal* is from Webster's Revised Unabridged Dictionary (1913):

> Goal 2. The final purpose or aim; the end to which a design tends...

This definition implies that designed objects have a purpose that is discernible from their form and intended environment: raindrops are aerodynamic and knives have sharp edges. Rain and knives have behaviors (falling, cutting, etc.) but I would not consider these to be intelligent behaviors. These objects have no options to behave differently, nor a mechanism to choose. I propose that the ability to choose a behavior with the aim to accomplish a goal is the critical feature of intelligence.

> Intelligence: Choosing actions to accomplish a goal.

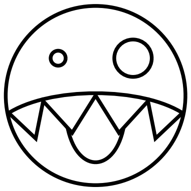
If a thing can make good choices I don't feel right calling it just a *thing*. I prefer to use the word **creature** to refer to any intelligent being, whether mechanically constructed or biologically evolved.

A Habitat for Intelligent Behavior

For experimentation and demonstration of intelligent processes, I have developed a tiny turn-based environment with two rules.

1. Every turn there is a 50% chance of a resource spawning nearby.
2. Every turn there is a 50% chance of a resource spawning at a farther distance.

I have also designed a creature to interact with this environment with two sensors and two actions.



- The little eye detects ***C***lose resources
- The big eye detects ***F***ar resources
- The teeth ***G***ather resources
- The tongue ***P***ulls far resources closer

Figure 1. The Anatomy of Bitey.

It's cute, isn't it? We are going to be working with this little guy so let's give it an endearing name from a classic TV show.

> I call the big one "Bitey."
> - Homer Simpson

My goal is for Bitey is to gather as many resources as it can. Objective sensors *C* and *F* will always read one of the two boolean values (0 or 1.) Actions *G* and *P* require a numeric parameter for execution to indicate the magnitude of the action.

A System of Behavior

If the core of intelligence is choosing an action, let's begin with a simple data structure and process to use it. Our data is a collection of behaviors, each consisting of a single **situation** (coinciding sensory information) and an action. The *situation* property is a comma separated string of the values of sensors *C* and *F* in order. The *action* property is a combination of the action, a colon, and a value for the action parameter.

```
behaviors = [
    {"situation": "0, 0", "action": "P: 0.0"},
    {"situation": "0, 1", "action": "P: 1.0"},
    {"situation": "1, 0", "action": "G: 3.0"},
    {"situation": "1, 1", "action": "G: 3.0"}
];
```

The process to utilize behavior data is:

1. Observe sensor values at the current point in time.
2. Find the situation column of the behavior that matches the current sensory information.
3. Perform the action from the matched behavior.
4. Repeat

Bitey can use this system to choose actions, but if it can't modify its own behavior table, does it really feel desire, or is it simply executing mindless instructions? I argue that even though it has a choosing process, and can therefore be considered to possess a rudimentary intelligence, if those choices can never be changed, its behavior is only mindless instinct. **Instinctual desire** for a goal does not truly reside in a creature, but in its creator. How could a different process allow Bitey to accept my goal as its own personal desire?

The Mathematics of Pleasure and Pain

C and *F* are objective senses. Their purpose is to measure the environment accurately, consistently, and quickly. These senses only report facts, not assessments of whether that information is good or bad. But if we want to give Bitey the ability to evaluate the benefit or harm caused by a behavior, that is exactly the opposite of what is needed. Instead of objective senses, we need **subjective senses** that don't care about the details of the environment, but whether what just happened was good or bad. But what is *good* and *bad* in a mathematical sense?

> I'm fuzzy on the whole "good/bad" thing.
> - Dr. Peter Venkman

I define an action to be **beneficial** (good) if it improves the odds of achieving a specified goal, and **harmful** (bad) if it decreases those odds. For example, subjective senses could return positive values when resources are acquired, and negative if Bitey wastes time and energy for no benefit. Subjective sensory data can be literally thought of as feelings of pleasure and pain.

The intensity of a subjective sense will be represented by a value between 1.0 and -1.0, inclusive.

- **1.0** is the maximum possible outcome. It should indicate that a creature's primary goal has been achieved and if it never accomplishes anything else, it can die happy.
- **0.0** indicates no effect on the chances of eventually accomplishing the primary goal.
- **-1.0** is the worst thing that could possibly happen: death and dismemberment, all previous accomplishments destroyed and other terrible disasters.

Let's have both of Bitey's actions return a subjective sensory response as a function of the magnitude of their action parameter.

```
actions.M.ss = function (param) {
    return -0.001 - param * 0.002;
};

actions.E.ss = function (param, situation) {
    if (situation[0] && param >= 3.0) {
        return param * -0.01 + 0.05;
    }
    return param * -0.01;
};
```

Action *P* appears to only return negative (painful) subjective sense values. Pulling costs time and energy and results in no immediate benefit. Bitey doesn't feel the simple joy of licking things, yet.

Action *G* is partially harmful (the parameter value multiplied by -0.01) but when the action parameter is greater than or equal to 3.0, it adds 0.05, resulting in a pleasurable response. Gathering resources expends a little time and energy, but when successful, can result in acquiring essential materials.

Blurry Action Parameters

If we don't allow Bitey to experiment outside of pre-programmed behaviors, it can't possibly ever discover better ones. Instead of deterministically performing action *G* with a parameter of 3.0, we could select from one of three parameter values: 2.0, 3.0, or 4.0. A simple **blurry action parameter** is a collection of parameter values, each with a likelihood of 0.0 to 1.0. Let's represent a collection of blurry parameters for action *G* for situation `1, 0` with this data structure.

```
bap.G["1, 0"] = [
    {"param": 2.0, "likelihood": 1.0},
    {"param": 3.0, "likelihood": 1.0},
    {"param": 4.0, "likelihood": 1.0},
];
```

We can calculate the absolute probability of a blurry parameter by dividing its likelihood by the sum of all likelihoods.

Bitey can now haphazardly choose a parameter, but making a random choice is not the same as making a choice with the aim of accomplishing a goal. To fix that, we will need to give it a way to compare the benefit or harm of different action parameters, then adjust likelihoods based on those results.

Blurry Parameter Tuning

There would be no point to feeling pain and pleasure if it couldn't change behavior. This can be achieved by adding the subjective sense result to the likelihood of the appropriate blurry parameter. If our creature is in situation `1, 0` and performs action *G* with a parameter of 3.0, it will result in a subjective sense of 0.02. This will affect the blurry parameter table for Action G/Situation [1, 0].

```
bap.G["1, 0"] = [
    {"param": 2.0, "likelihood": 1.0},
    {"param": 3.0, "likelihood": 1.02},
    {"param": 4.0, "likelihood": 1.0},
];
```

Because the subjective sensory data for action *G* with a parameter of 3.0 felt good (was greater than 0.0) it is slightly more likely to be chosen next time. (Up to 33.8% from 33.3%.)

Blurry Parameter Normalization

To keep the amount of change from subjective senses consistent, blurry parameter likelihoods should be divided by the highest likelihood so the maximum value is always adjusted to 1.0.

```
bap.Y["1, 0"] = [
    {"param": 2.0, "likelihood": 0.98},
    {"param": 3.0, "likelihood": 1.0},
    {"param": 4.0, "likelihood": 0.98},
];
```

Through the effects of tuning and normalization some parameter values may become highly unlikely. After normalization, we could also clean up insignificant likelihoods by removing parameters beneath a specified threshold. The tuning process should slowly reduce multiple blurry parameters to only the most beneficial or least harmful. More advanced logic could allow adding new blurry parameters to explore outside of those initially programmed and dividing a single parameter into multiples for increased accuracy.

Action Effects

Actions should be able to manipulate the environment, but Bitey's have no effects. Let's give it the ability to alter the upcoming turn's resource probabilities. We'll have action *P* make spawning a nearby resource on the next turn five times more likely if sensor *F* is 1 and action *P* has a parameter is greater than or equal to 1.0.

```
actions.P.effect = function (situation, param) {
    if (situation[1] && param >= 1.0) {
        return [5.0, 1.0];
    }
    return [1.0, 1.0];
};
```

Action *G* will increase the chances that the next turn will spawn a nearby resource if it was executed with an action parameter greater than 3.0.

```
actions.G.effect = function (situation, param) {
    if (situation[0]) {
        if (situation[1] && param > 3.0) {
            return [5.0, 1.0];
        }
    }
    return [1.0, 1.0];
};
```

Bitey now has the power to improve its future situations. Unfortunately, its subjective senses can only evaluate what is happening in the current situation so there is no way for it to judge if these longer-term effects will result in harm or benefit.

Virtual Subjective Senses

To give Bitey some wisdom about the long-term consequences of its actions, we can add a process to remember subjective senses that were felt at the same time as situations were observed. This can upgrade objective senses *C* and *F* to also return subjective sensory feelings.

> I've got a bad feeling about this.
> - Han Solo, et al.

Below is a simple data structure for virtual subjective senses, or as I also refer to them, **opinions**.

```
opinions = [
    {"situation": "0, 0", "vss": 0.0},
    {"situation": "0, 1", "vss": 0.0},
    {"situation": "1, 0", "vss": 0.0},
    {"situation": "1, 1", "vss": 0.0}
];
```

An opinion has an initial virtual subjective sense (vss) value of 0, then after every action, it is updated to the average of the current subjective sense plus the previous value for the appropriate situation.

```
vss.forEach(function (opinion, i, a) {
    if (current_situation === opinion.situation) {
        a[i].vss = (opinion.vss + current_subjective_sense) / 2;
    }
});
```

For example, if situation `1, 1` results in a subjective sensory value of 0.02, we should update the vss of the appropriate opinion to (0.02 + 0.0) / 2, or 0.01 like so:

```
opinions = [
    {"situation": "0, 0", "vss": 0.0},
    {"situation": "0, 1", "vss": 0.0},
    {"situation": "1, 0", "vss": 0.0},
    {"situation": "1, 1", "vss": 0.01}
];
```

Subjective Sense Aggregation

If we want to combine true subjective sense values with virtual ones, we will need an appropriate way to combine them. Adding them together could result in intensities greater than 1.0 or less than -1.0, inconsistent with the definition of subjective sense intensities as described above. Let's create an aggregation rule that makes more sense.

First, separate the positive and negative values into collections of pleasurable and painful feelings, respectively. Then sort each by absolute value, descending.

```
pleasure.sort(function (a, b) {
    return b - a;
});

pain.sort(function (a, b) {
    return Math.abs(b) - Math.abs(a);
});
```

Each value should be added together one at a time, in order, weighted by the difference between the running total and 1.0 (the maximum value.)

```
pleasure.reduce(function (acc, current_value) {
    return current_value * (1.0 - acc) + acc;
}, 0.0);

pain.reduce(function (acc, current_value) {
    return Math.abs(current_value) * (1.0 - acc) + acc;
}, 0.0);
```

The final value is simply the aggregate pleasure minus the aggregate pain.

```
feelings = pleasure - pain;
```

This should result in a sensible overall subjective sensory value even when combining many different indictors of benefit and harm. One could also infer the riskiness of a behavior from the difference between the pleasure and pain aggregates.

The Process of Emotional Desire

Let's gather all of our data and processes into a procedure to clarify how they work together.

1. Perform the action in the behavior table indicated by the current situation.
2. Add subjective sense values of that action into the current subjective sense state.
3. Wait to observe the resulting situation.
4. Add the virtual subjective sense value associated with the consequent situation to the subjective sense collection.
5. Reduce the subjective sense state to a single aggregate value.
6. Tune and normalize the blurry action parameters associated with the previous situation.
7. Update the virtual subjective senses for the previous situation.
8. Clear the current subjective sense state.
9. Repeat.

Using this process, Bitey can alter its behavior toward feelings of pleasure and away from feelings of pain. It can even have forward-looking behavior after learning to recognize harmful and beneficial situations.

Conclusion

But so what? Why should poor Bitey have to experience the joy and suffering of emotional desire when it would be simpler to program it to execute what we deem are optimal behaviors? In the above examples, there is no advantage. In fact, blurry action parameters make the quality of its behavior much worse. An optimally programmed creature doesn't need to waste time and energy trying out lousy action parameters. But what if the environment changed? What if gathering became slightly more difficult and required an action parameter of 3.1 for success? Our rigidly programmed automaton would ignorantly fail, and continue to fail, until its creator was forced to intervene. But an emotional creature possesses feelings of success and failure and has been given the reins to adjust its own behavior. Flexible behavior systems grant emotional creatures the ability to thrive in multiple and unpredictably changing environments.

Addendum: Conscious Desire

Creatures with emotional desire can feel, but cannot truly think about their situation. What would Bitey need to consciously envision its goal? With a firm base of emotional desire, surprisingly little else would be required; only predictive situation data and a process for its use. The new data would be a simple collection of consequent situations resulting from a specific initial situation and action. The emotional desire process already handles this information when dealing with virtual subjective senses. Once it has sufficient understanding of how its behavior affects the environment, it can imagine the consequences of its actions and evaluate imaginary situations with its opinions to decide the best plan of action. I am continuing to research and develop this type of system, but alas, I have already expended an appropriate amount of time and words for this essay.

External Resources

Companion site with working code examples

https://chrisbroski.github.io/goal-math

The public repository https://github.com/chrisbroski/goal-math is a work in progress and will continue to be improved as time allows.

The Foundations of Behavioral Logic

http://behaviorallogic.com/foundation

This site documents my hypotheses of intelligence, which this essay is based upon.