Elliptic Paraboloid Machine Learning

by Sven Nilsen, 2017

In this paper I represent an idea of how to create artificial intelligence that learns to make trade-offs in environments with a known unknown utility function. This is a step closer toward formalization of zen rationality: The ability to reason and make optimal decisions toward known unknown goals.

The basic principle of elliptic paraboloid machine learning is to replace the utility function with a higher order utility function with known unknown constants. This function takes the form of an N-dimensional elliptic paraboloid, which has the following important mathematical properties:

1. The function is convex, making it efficient to optimize.
2. When unconstrained, each pair of variables has a trade-off ratio that only depends on the pair.

This means by observing examples of trade-offs being made in the real world, the AI can learn the utility function gradually by deriving the true value of the unknown constants.

A N-dimensional elliptic paraboloid has the following function:

$$\text{elliptic\_paraboloid} := \backslash(a : [\text{real}],\ k : [\text{real}]) = \backslash(x : [\text{real}]) = \sum i\ \{\ a_i^2 \cdot (x_i - k_i)^2\ \}$$

Notice that the function is constructed from a list of positive constants `a` and `k`. The `a` constants describe how much weight each sub-goal has. The `k` constant describe what the sub-goals are.

The values `x` are measured from the model of the world, simulated by the AI as consequence of taking various actions. When the AI observes that one action is preferable to another, it guesses which pair of variables that determines the preference. It knows that there is a trade-off between the two variables in a given situation where neither action is preferable to another, and this trade-off is independent of all other variables, so the observed action puts a constraint on the true trade-off. This constraint also depends on the sub-goal. The AI learns both the trade-offs and the sub-goals from observations.

For example, I am training the AI to clean up my room. On the floor there are two shoes, which I put on the shelf. The AI observers that the number of shoes on the floor was 2, which I reduced to 0, so it infers that the preferred number of shoes on the floor should be 0. It also observes that the number of shoes on the shelf was 0, but I increased to 2, so it infers that the preferred number of shoes on the shelf should be 2. My utility function so far (to be minimized):

$$a_{floor}^2 \cdot (x_{floor} - 0)^2 + a_{shelf}^2 \cdot (x_{shelf} - 2)^2$$

What the AI does not know yet, is how much I prefer there to be shoes on the shelf versus how much I prefer there to be no shoes on the floor. There is a trade-off, depending on the number of shoes on the floor and the shelf, where increasing the number of shoes on the shelf with one, corresponds to decreasing the number of shoes on the floor with some unknown value. The utility of shoes on the floor can be traded with the utility of shoes on the shelf, but the market value fluctuates, depending on where the shoes are, but only shoes and nothing else. When confronted with a dilemma, where the AI must choose either to put shoes on the shelf or take away shoes from the floor, it learns the ratio `$a_{floor} / a_{shelf}$`.