# Approximating optimal PRs by approximating the VOC

If we can approximate the VOC, then it would be easy to approximate the expected value of the next belief state, because

This also gives us an approximate of the meta-level Q-function of the current state:

The optimal state-value function can be written as

We can therefore express the optimal pseudo-reward as

And the optimal integrated pseudo-reward becomes

Therefore, if we can approximate the VOC we could apply this VOC approximation to approximate the optimal PRs directly.

The VOC might be easier to approximate than the optimal value function because it is only about the change whereas the optimal value function has two components to it: the current value and the change. Another advantage of this approximation is that it eschews having to approximate the expected value of the state-value function across potentially infinitely many possible next states.

There are two kinds of approximations that we can make to the VOC:

1. Extrapolation from n-step VOC
2. Fitting the VOC using a number of features, including n-step VOCs, but also regret reduction, and uncertainty reduction
3. Blinkered VOC
4. Gaussian processes, support vector regression, and other advanced machine learning techniques for non-linear regression.

The first two types of approximation can also be combined with the third one.

## Linear approximation

We found that in the case of sampling from a normal distribution, the optimal meta-level state-value function can be efficiently approximated by

where the value of perfect information (VPI) is defined by

where is the action that currently appears best and is the action that currently appears second best.



## Blinkered approximation

The blinkered approximation can be very efficient in problems with many possible object-level actions, because the number of possible computation sequences is . To temper this combinatorial explosion, we decompose the problem into one meta-level MDP for each object level action that is currently available (cf. Hay et al., 2012). In this way, the base is where is the set of computations that evaluate object-level action . This effectively reduces the the exponent by a factor of .

This greatly reduces the computational complexity of the meta-planning problem. For our concrete example, we will have 4 meta-level MDPs with only 3 possible computations each. This is very tractable! I want to implement it!

## Test cases

|  |  |  |
| --- | --- | --- |
|  | **two actions** | **Many actions** |
| **Two outcomes** | one-lightbulb problem | Hay’s n-armed bandit meta-MDP |
| **Many outcomes** | reasoning about the EV of a continuous payoff | Mouselab-MDP |