Notes on Powell’s Unified Modeling Framework

Compiled by Dimitar Gueorguiev, 12/26/2024

The core elements of the universal modeling framework are:

**1 .** *State variables* – the state variable contains everything we know and only what we need to know to make a decision and model our problem. State variables include physical state variables (e.g. the location of a drone, inventories, investments in stocks), other information about parameters and quantities we know perfectly (such as current prices and weather), and beliefs in the form of probability distributions describing parameters and quantities we do not know perfectly (e.g. an estimate of how much a drug will lower the blood sugar in a new patient, or how the market will respond to price).

**2 .** *Decision variables* – a decision variable can be binary (e.g. hold or sell), a discrete set (e.g. drugs, products, paths), continuous variables. Decisions are subject to constraints and we make decisions using a method we call *a policy* .

**3.** *Exogenous information* – this is the information that we learn after we make a decision (e.g. market response to a price, patient response to a drug, the time to traverse a path), that we did not know when we made the decision. Exogeneous information comes from outside the system we are modeling.

Note: Decisions, from other side, can be thought as an *endogenous information* since we make those decisions, which represent information internal to the process.

**4.** *Transition Function* which consists of the equations required to update each element of the state variable . This covers all the dynamics of our system, including the updating of estimates and beliefs for sequential learning problems.

**5.** *Objective Function* – can be expressed as contribution in terms of reward or cost we make each time period, given by where is determined by our policy, and is the current state, computed by the transition function.

The most common way to express and maximize the objective function is to maximize the cumulative contributions of a reward metric

(1)

where the expectation notation basically means “take an advantage over all types of uncertainty”. This could be the uncertainty how a drug will perform, or how market will respond to price (captured in the initial state ), as well as the uncertainty in the information that arrives over time.

The maximization over policies simply means that we want to find the best method for making decisions.

Once we have identified these five components, we still have two remaining steps to complete:

**6.** *Stochastic modeling* (aka *Uncertainty quantification*)- There can be uncertainty about parameters and quantities in the state variable (including the initial state ), as well as our exogenous information process . In some instances we may avoid modeling the process by observing a physical system. Otherwise, we need a mathematical model of the possible realizations of given and our decision . Note that both and can influence .

**7.** *Designing policies* – only after we are done with modeling, we turn to designing the policy .

The policy consists of some type of function , possibly with tunable parameters that are associated with the function f, where the policy maps the state to a decision. The policy will often contain an imbedded optimization problem within the function. This means that we can write (1) as

(2)

Open Question: How do we search over functions?

After this characterization we can represent the decision problem which this modeling framework addresses as a sequence (infinite or finite) with the following structure:

Notice the presence of the following repetitive pattern in the sequence – this pattern encodes the sequential steps “state, decision, new information” and it captures what we know (the state ), which we use to make a decision (which earns reward or incurs a cost) , where the decision comes from a policy .

There are many problems where it is more natural to use a counter (the -th experiment, the -th customer arrival), in which case we would write our sequential decision problem as

In the most general settings we will see the sequence

Note that or can be infinite or finite. So we can have both finite pattern “*state, decision, information, state, decision, stop*” as well as “*state, information, decision, state, information, decision, state, …*”. When or is infinite, we say the problem has *infinite horizon*.

# Illustration Example for the Unified Modeling Framework (UMF)

The Inventory problem

The Inventory Problem is one of the simplest and most classical sequential decision problems. We want to manage the inventory of a product to serve demands over time. Let be our inventory at time , is how much we order (that arrives instantly), to serve a demand that is not known at time . The evolution of the inventory is given by :

(3)

One possible kind of policy for the inventory problem is: when the inventory falls below a value , order enough to bring it up to . In order to select a specific policy of this kind we need to determine the parameter vector .

The state in this problem is the inventory . The decision variables in this problem is simply what we order at time , assuming for simplicity that it arrives right away. We also introduce our policy , where which we will design after we create our model.

Exogeneous information is the demand that arises between and .

The transition function would be the evolution of our inventory , given by

(4)

The objective function in this problem is an instance where it is more natural to write the single-period contribution function *after* we observe the information since this contains the demand that we will serve with the inventory we order in period . Thus we can write our contribution function as

where is the price at which we sell our product, and is the cost per unit of product. Our objective function would be given with

where and we have to be given a model of the exogenous information process . Since the exogenous information is random, we have to take the expectation of the sum of contributions to average over all possible outcomes of the information process.

We also need to develop a mathematical model of the distribution of demand .

We already communicated than one popular policy class is the one that has an order-up-to structure given by

(5)

This is a parametrized policy, which leaves the challenge of finding the parameter vector by solving

(6)

So we have chosen a specific class of policy (5) and in (6) we optimize over this class to find the best instance of this class which suits our needs.

# Designing Policies for Sequential Decision Problems

Making decisions under uncertainty can be organized along two broad strategies for creating policies:

i ) *Policy Search* – this includes all policies where we need to search over.

Different classes of functions for making decisions. For example the order-up-to policy in (5) is a form of nonlinear parametric function. The parametric functions ( for example) introduce some tunable set of parameters . We find this set of parameters by maximizing suitably defined objective function (e.g. (6)).

ii ) *Lookahead approximations* – these policies are formed in order to make the best decision at the current moment given an approximation of the downstream impact of the decision. There are policy classes associated with these lookahead approximations.

Each of these two strategies produce policies that can be divided into two classes, creating four classes of policies. We describe these policy classes below.

## Policy Search

Policies in the policy search class can be divided into two subclasses:

1 ) *Policy function approximations* (PFAs)-

These are analytical functions that map a state (which includes all the information available to us) to a decision (the order-up-to policy in (5) is a PFA). These analytical functions come in three flavors:

*Lookup tables*-

These are used when a discrete state can be mapped to a discrete action.

*Parametric functions-*

These describe any analytical functions parametrized by a vector of parameters .

For example, consider the order-up-to policy in the Inventory Problem. We can write it as a linear model as

where are features extracted from information in the state variable.

*Nonparametric functions*-

These include functions that might be locally linear approximations based on neural nets for instance.

2 ) *Cost function approximations* (CFAs)-

CFA policies are parameterized optimization models (typically deterministic optimization models) that have been modified to help them respond better over time, and under uncertainty. CFAs have an imbedded optimization problem within the policy.

Example: A simple CFA used in learning problems is called *interval estimation* and might be used to determine which ad gets the most clicks on a website. Let be the set of ads and let be our current best estimate of the probability that ad will be clicked on after we have run observations (across all ads). Then let be the standard deviation of the estimate . Interval estimation would choose as the next ad using the policy

*Interval estimation* CFA would choose as the next ad using the policy

(7)

Note that solving an optimization problem within a policy (e.g. (7)) implies that we can solve *any* parametrized optimization problem. We are no longer restricted to the idea that x has to belong to a finite set; it can be a large integer program, such as those used to plan transportation schedules with schedule slack inserted to handle possible weather delays, or planning integer program planning resource consumption for the next time interval with reserves in case the resource producing facility fails.

## Policies based on Lookahead Approximations

A natural strategy for making decisions is to consider the downstream impact of the decision made in the present moment. There are two ways of accomplishing this: