Joe Large
CS497
Notes for 2/11 and 2/13

1 Sequential Games Against Nature

We will start by assuming that the states are always known (i.e. can be perfectly measured). As usual, we will let $u_k \in U(x)$ denote the actions of the decision maker, and $\theta_k \in \Theta$ denote the actions of nature. And finally, we will assume that nature is *evil* (i.e. non-deterministic).
Define the *state transition equation* as $x_{k+1} = f(x_k, u_k, \theta_k)$.
Define the *loss functional* as $L = l_I(x_1) + \sum_{i=1}^{k} l(x_i, u_i, \theta_i) + l_F(x_F)$.

1.1 Nondeterministic Approach

In this approach, we assume the worst case. The following would be a **one-stage forward projection**:

$$F(x_k, u_k) = \{x_{k+1} | \exists \theta_k \in \Theta(x_k, u_k) \ for \ which \ x_{k+1} = f(x_k, u_k, \theta_k)\} \subseteq X$$

A **two-stage forward projection** would be of the following form: $F(F(x_k, u_k), u_{k+1})$.
So, in the above equations, we are in state x_k and we are thinking about applying action u_k. Then, these forward projections give us a multi-branch tree which gives us all the possible paths.

1.2 Probabilistic Approach

In this approach, we try to determine the probability of the different actions that nature could possibly take based on the information that we have. So, we define the following:
$P(\theta_k | x_1, \ldots, x_k, u_1, \ldots, u_k, \theta_1, \ldots, \theta_{k-1})$
In order to reduce to a simple algorithm, we will use a *Markovian assumption* and say that the above probability is equal to $\mathbf{P}(\theta_\mathbf{k} | \mathbf{x_k})$. Then by using $P(\theta_k | x_k)$ along with the *state transition equation* ($x_{k+1} = f(x_k, u_k, \theta_k)$), we get $\mathbf{P}(\mathbf{x_{k+1}} | \mathbf{x_k}, \mathbf{u_k})$
The above is a **one-stage probabilistic forward projection**.

A **two-stage probabilistic forward projection** would be
$P(x_{k+2}|x_k, u_k, u_{k+1}) = \sum_{x_{k+1}} P(x_{k+2}|x_{k+1}, u_{k+1}) \cdot P(x_{k+1}|x_k, u_k)$
We get this equation by using the formula $P(A|C) = \sum_B P(A|B) \cdot P(B|C)$

2 Strategy / Conditional Planning

The *strategy function* maps states to actions.
$\gamma : X \to U \qquad u_k = \gamma(x_k)$
It is important to note the following points:

- We can now search the strategy space for actions.
- If nature is interfering at every step, what happens when we fix γ ?
- This is a form of feedback control (or reactive planning).

Using the nondeterministic approach, we can use a table to keep track of all states reachable from the current state by applying the given strategy.
In the probabilistic case, we can keep the probabilities in the table to record what are the chances of reaching a certain state given the current state. Now, we will use a parameterized Markov Chain to get $P(x_{k+1}|x_k = 1, u_k = \gamma(1))$.
In the extreme case of the Markov Chain, you could pick the strategy to always be the same action. So, the *key idea* is that we are looking to choose a strategy that minimizes the **worst case cost** in the nondeterministic cost, or choose a strategy that minimizes the **expected cost** in the probabilistic case.

2.1 BDP: Nondeterministic Cost-to-go

Here, we need to consider the worst thing that nature could do, and then pick the action that minimizes the loss.
$L^{\star}_{1,F}(x_1) = min_{u_1}(max_{\theta_1} \ldots (min_{u_k}(max_{\theta_k}\{l_I(x_1) + \sum_{i=1}^{K} l(x_i, u_i, \theta_i) + l_F(x_F)\})))$
Unfortunately, this approach develops into a very large min-max problem that could require up to exponential time to solve. So, perhaps there is an easier solution.

2.2 BDP: Probabilistic Cost-to-go

In this case, we are simply minimizing the expected loss functionals through minimizing the action selected.

$L^\star_{1,F}(x_1) = min_{u_1,\ldots,u_k}\{E_{\theta_1,\ldots,\theta_k}[l_I(x_1) + \sum_{i=1}^{K} l(x_i, u_i, \theta_i) + l_F(x_F)]\}$

2.3 DP: Nondeterministic Cost-to-come

Now, the challenge is to express $L^\star_{k,F}$ in terms of $L^\star_{k+1,F}$. In the nondeterministic case, we end up with the following equation (through induction on $L_{F,F}$):

$L^\star_{k,F}(x_k) = min_{u_k \in U(x_k)} max_{\theta_k \in \Theta(x_k,u_k)}\{L^\star_{k+1,F}(x_{k+1}) + l(x_k, u_k, \theta_k)\}$

where $x_{k+1} = f(x_k, u_k, \theta_k)$

2.4 DP: Probabilistic Cost-to-come

This case reduces to the following equation:

$L^\star_{k,F}(x_k) = min_{u_k \in U(x_k)}\{l(x_k, u_k) + \sum_{x_{k+1}} L^\star_{k+1,F}(x_{k+1}) \cdot P(x_{k+1}|x_k, u_k)\}$

3 Cycle Problems

For nondeterministic problems, there must be no negative cycles, and you must be able to escape positive loss cycles.

For probabilistic problems, it is a little easier since there is usually some probability (however small) that you will eventually escape the cycle. But, we still must ensure that there are no negative cycles with probability 1 (since there would no chance to leave the cycle), and you must be able to escape a positive loss cycle with probability 1.