CS497 Planning and Decision Making Week 4 Notes (Feb. 11, 13)

Scribe: Warren Shen

Outline

non-deterministic forward projection
probabilistic forward projection
non-deterministic dynamic programming
probabilistic dynamic programming
cycles and termination

1 Sequential Games Against Nature

Notation

$U(x_k)$	the set of decision maker actions from the state x_k
$u_k \in U(x_k)$	
$\Theta(x_k)$	the set of actions nature can perform in state x_k
$\theta_k \in \Theta(x_k)$	
$\theta_k \in \Theta(x_k, u_k)$	like above, except that nature responds to decision maker

The state transition equation: $x_{k+1} = f(x_k, u_k, \theta_k)$
The cost functional: $L = \sum_{i=1}^{K} l(x_i, u_i, \theta_i) + l_F(x_F)$

Use termination actions as before. Also, assume the current state is always known.

1.1 Non-Deterministic Forward Projection

If nature is non-deterministic, what will our next state be given our current state and our action that we apply?

Let θ_k be whatever action nature does after we apply u_k in state x_k, and $F(x_k, u_k)$ be the set of states that we can be in after u_k and θ_k is applied when we were in state x_k. So, we have $\theta_k \in \Theta(x_k, u_k)$ and $F(x_k, u_k) \subseteq X$, where X is the set of all states.

In the non-deterministic case, we have

$$f(x_k, u_k) = \{x_{k+1} \mid \exists \theta_k \in \Theta(x_k, u_k) \text{ such that } x_{k+1} = f(x_k, u_k, \theta_k)\}$$

This is a 1-stage forward projection, where we project all possible states one step forward. Here is a 2-stage forward projection:

$$F(F(x_k, u_k), u_{k+1})$$

This can be further expanded to any number of stages.

1.2 Probabilistic Forward Projection

In the probabilistic case, assume we have a probability distribution over the possible actions nature can do given our action u_k in state x_k. Thus, the probability that nature performs action θ_k could be written as

$$P(\theta_k \mid x_1 \cdots x_k, u_1, \cdots u_k, \theta_1 \cdots \theta_{k-1})$$

which is the probability given everything that has happened in the past.

This is too big - so our solution is to arbitrarily knock out stuff until we're happy. To make it not so arbitrary, we'll go by the Markovian assumption and say that the probability depends only on local values. So, now we have

$$P(\theta_k \mid x_k)$$

Now, given this probability as well as the fact that we're in state x_k and apply action u_k, we want to get the probability that we'll get to state x_{k+1}. We can simply combine the state transition function, $x_{k+1} = f(x_k, u_k, \theta_k)$, and $P(\theta_k, \mid x_k)$ to get $P(x_{k+1} \mid x_k, u_k)$. This is a 1-state probabiltic forward projection.

A 2-stage probabilistic forward projection is the probability that we'll get to a state x_{k+2} from x_k. This probability is $P(x_{k+2} \mid x_k, u_k, u_{k+1})$. In order to get this to a form we know, we can marginalize variables and get

$$P(x_{k+2} \mid x_k, u_k, u_{k+1}) = \sum_{x_{k+1}} P(x_{k+2} \mid x_{k+1}, u_{k+1}) P(x_{k+1} \mid x_k, u_k)$$

1.3 Strategies

A strategy $\gamma : x \to u$ is a plan that tells the decision maker what action to take given a state $(u_k = \gamma(x_k))$. Remember that our c ost functional is $L = l_i(x_i) + \sum_{i=1}^{k} l_(x_i, u_i, \theta_i) + l_f(x_f)$. How should we choose our strategy?

In the non-deterministic case, we can think of our strategy problem as a table, where each row specifies a state we could be in right now, and each column specifies states we could go to if we applied $\gamma(x_i)$. Wherever there is a 1, that means that the state is reachable from our current state.

	1	2	3	.	.	.	n
1	0	1	1	.	.	.	1
2	1	0	1	.	.	.	1
3	0	1	0	.	.	.	0
.							
.							
.							
n	0	0	1	.	.	.	0

For example, if we were in state 2 and we applied the strategy that this table represents, we could end up in $1, 3...n$, but not 2 because there is a 0 in that position. Since nature is non-deterministic, we don't know which state we'll end up in.

So, given our cost functional L, we want to choose γ to minimize the worst-case cost.

In the probabilistic case, we can also think of our strategy in terms of a table, except that instead of having boolean values in the entries, we have probabilities:

	1	2	3	.	.	.	.n
1	.6	.02	.01	.	.	.	.3
2	.02	.4	.2	.	.	.	.13
3	.2	.4	.02	.	.	.	.2
.							
.							
.							
n	0.0	.14	.33	.	.	.	.43

For example, in the entry $(2,3)$, we have $P(x_{k+1} = 3 \mid x_k = 2, u_k = \gamma(2)) = .2$.

Given our cost functional L, in the probabilistic case, we want to choose γ to minimize the expected cost.

1.4 Dynamic Programming in Games against Nature

1.4.1 Setting it up

As before, we have stages $1, 2, \cdots, K-1, K, F$. Here are some definitions, which have been adapted from our dynamic programming for games without nature:

In the non-deterministic case:

$$\begin{aligned} L^*_{F,F}(x_F) &= l_f(x_f) \\ L^*_{K,F}(x_K) &= \min_{u_K} \max_{\theta_K} \{l(x_K, u_K, \theta_K) + l_F(x_F)\} \\ L^*_{1,F}(x_1) &= \min_{u_1} \max_{\theta_1} \min_{u_2} \max_{\theta_2} \cdots \min_{u_K} \max_{\theta_K} \left\{ l_I(x_1) + \sum_{i=1}^{K} l(x_i, u_i, \theta_i) + l_F(x_F) \right\} \end{aligned}$$

In the probabilistic case:

$$\begin{aligned} L^*_{F,F}(x_F) &= l_f(x_f) \\ L^*_{K,F}(x_K) &= \min_{u_K} \{E_{\theta_K} [l(x_K, u_K, \theta_K) + l_F(x_F)]\} \\ L^*_{1,F}(x_1) &= \min_{u_1 \cdots u_k} \left\{ E_{\theta_1 \cdots \theta_K} \left[l_I(x_1) + \sum_{i=1}^{K} l(x_i, u_i, \theta_i) + l_F(x_F) \right] \right\} \end{aligned}$$

where E_{θ_K} is the expectation of θ_K, and $E_{\theta_1 \cdots \theta_K}$ is the expectation over $\theta_1 \cdots \theta_K$

1.4.2 Finding the minimum loss

Now, to find the loss from any stage k, we use dynamic programming, as before. In the non-deterministic case we have

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

where x_{k+1} is defined by our state transition equation $x_{k+1} = f(x_k, u_k, \theta_k)$.

In the probabilistic case we have

$$\begin{aligned} L^*_{k,F}(x_k) &= \min_{u_k \in U(x_k)} \left\{ E_{\theta_k} \left[L^*_{k+1,F}(x_{k+1}) + l(x_k, u_k, \theta_k) \right] \right\} \\ &= \min_{u_k \in U(x_k)} \left\{ \sum_{\theta_k} (L^*_{k+1,F}(x_{k+1}) + l(x_k, u_k, \theta_k)) P(\theta_k \mid x_k, u_k) \right\} \end{aligned}$$

However, if $l(x_k, u_k, \theta_k) = l(x_k, u_k)$, then our formula becomes

$$L^*_{k,F}(x_k) = \min_{u_k \in U(x_k)} \left\{ l(x_k, u_k) + \sum_{x_{k+1}} (L^*_{k+1,F}(x_{k+1})) P(x_{k+1} \mid x_k, u_k) \right\}$$

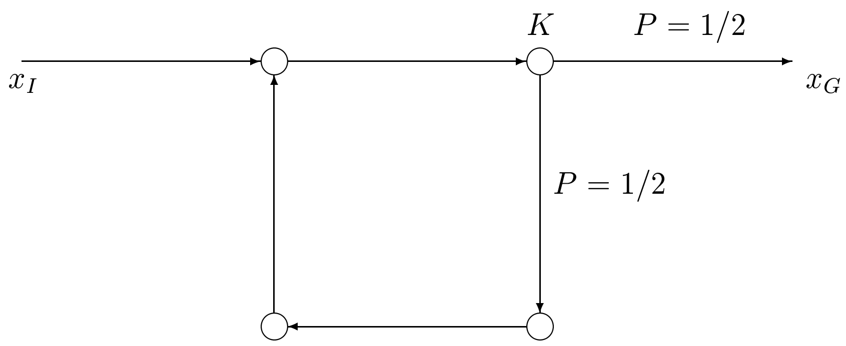
Here, we've just made the θ go away, but in reality it's just hiding inside of $P(x_{k+1} \mid x_k, u_k)$.

1.5 Cycles

As before assume termination actions to end our game when we reach a goal state, and also assume that K is unknown. What if there are cycles in our problem, where a series of actions could potentially bring you back to the same state? How do we make sure that our program terminates?

In the non-deterministic case, there must be no negative cycles (in reality, there must be no minimax cycles), and there also must be a way to escape or avoid all positive cycles. If there were negative cycles, meaning that even with nature we can perform actions in a cycle such that we still have negative loss, then the optimal strategy would be to go around forever to minimize loss. If there were positive cycles that we couldn't escape from or avoid, then it would be possible for nature to keep on sending us through the cycle forever.

In the probabilistic case, as long as no transitions at the start of a cycle is 1, then we will terminate. For example, suppose we had this graph, where nodes are states and edges are transitions:



Assume that all transitions have a loss of 1. If we were at state K, we would want to go to x_G. If we go straight from x_1 to x_g, we we'll have a total loss of 3. However, at K

there is a chance of 1/2 that we will go around the cycle. If we go around, we'll acquire an additional loss of 4 each time. Thus, the expected loss becomes

$$\begin{aligned} E[L] &= \frac{1}{2}(3) + \frac{1}{4}(7) + \cdots \\ &= 3 + \sum_{i=0}^{\infty} \frac{1}{2^{i+1}}(4i) \\ &< \infty \end{aligned}$$

If the probabilities are less than 1, then the expected loss converges on some finite value, meaning that we will terminate. However, no matter how far we go in the future, there will be some exponentially small chance that we will keep going around the cycle. To calculate L_k^*, when do we stop? We can pick some threshold ϵ and terminate when $\max_{x \in X} \left| L_{k+1}^*(x) - L_k^*(x) \right| < \epsilon$.