Advanced Predictive Modeling

Dr. Bowen Hui
Computer Science
University of British Columbia Okanagan

Additional MDP Example

- Suppose we have a machine component at hand, and we have to decide what to do with it.
- If the component is performing well, it will create quality products properly. However, if the performance is mediocre, or low, the products that get created will be of low quality and cannot be used. The lower the performance of the component, the more such bad products it creates.
- As the component gets older, its performance degrades.
- At some point, when there is not much “life” left in the component, we need to replace it with a new one. However, replacing a component comes with a high cost. Alternatively, we could repair it. This has a lower cost, and only boosts performance slightly.

States

- Let’s keep the model simple, and define two feature variables:
  - Age:
    - Age of the component
    - Values: old, middle, new
  - Perf:
    - Performance of the component
    - Values: poor, passing, good, excellent
- States:
  - Take all combinations of the two features to get a total of 12 states

Actions

- Replace
  - Replace with brand new component
  - High cost needed each time
  - Age of component returns to brand new
  - Performance of component becomes excellent
- Fix
  - Fix component
  - Small cost needed each time
  - Age of component does not change
  - Performance of component increases one level
- None
  - Let it run as is (do nothing)
  - No cost
  - No change to age or performance

Transition Function
$Pr(S_t | S_{t-1}, A=Replace)$

S_{t-1}		S_t											
		poor	poor	poor	pass	pass	pass	good	good	good	excel	excel	excel
		old	mid	new	old	mid	new	old	mid	new	old	mid	new
poor	old	0	0	0	0	0	0	0	0	0.1	0	0	0.9
poor	mid	0	0	0	0	0	0	0	0	0.1	0	0	0.9
poor	new	0	0	0	0	0	0	0	0	0.1	0	0	0.9
pass	old	0	0	0	0	0	0	0	0	0.1	0	0	0.9
pass	mid	0	0	0	0	0	0	0	0	0.1	0	0	0.9
pass	new	0	0	0	0	0	0	0	0	0.1	0	0	0.9
good	old	0	0	0	0	0	0	0	0	0.1	0	0	0.9
good	mid	0	0	0	0	0	0	0	0	0.1	0	0	0.9
good	new	0	0	0	0	0	0	0	0	0.1	0	0	0.9
excel	old	0	0	0	0	0	0	0	0	0.1	0	0	0.9
excel	mid	0	0	0	0	0	0	0	0	0.1	0	0	0.9
excel	new	0	0	0	0	0	0	0	0	0.1	0	0	0.9

Intuition for $Pr(S_t | S_{t-1}, A=Replace)$

- No matter what the current state of the component is, once we replace it, it will become new
- From a good company, we can expect that the performance of a new component will generally be excellent, with a small chance of it being good

- Note: If you want to indicate that all new components will be excellent, then there will be 1's in the last column and 0's everywhere else

Transition Function

$Pr(S_t | S_{t-1}, A=Fix)$

St_{-1}		S_t											
		poor	poor	poor	pass	pass	pass	good	good	good	excel	excel	excel
		old	mid	new	old	mid	new	old	mid	new	old	mid	new
poor	old	0	0	0	1	0	0	0	0	0	0	0	0
poor	mid	0	0	0	0	1	0	0	0	0	0	0	0
poor	new	0	0	0	0	0	1	0	0	0	0	0	0
pass	old	0	0	0	0	0	0	1	0	0	0	0	0
pass	mid	0	0	0	0	0	0	0	1	0	0	0	0
pass	new	0	0	0	0	0	0	0	0	1	0	0	0
good	old	0	0	0	0	0	0	0	0	0	1	0	0
good	mid	0	0	0	0	0	0	0	0	0	0	1	0
good	new	0	0	0	0	0	0	0	0	0	0	0	1
excel	old	0	0	0	0	0	0	0	0	0	1	0	0
excel	mid	0	0	0	0	0	0	0	0	0	0	1	0
excel	new	0	0	0	0	0	0	0	0	0	0	0	1

Intuition for $Pr(S_t | S_{t-1}, A=Fix)$

- When a component is fixed, its age stays the same, but performance will improve one level from what it used to be
- If the performance is already excellent, then it just stays as excellent after repair

- Note: you could potentially increase the level of uncertainty by saying that performance may increase more than 1 level or may stay the same

Transition Function
$Pr(S_t | S_{t-1}, A=None)$

St_{-1}		S_t											
		poor	poor	poor	pass	pass	pass	good	good	good	excel	excel	excel
		old	mid	new	old	mid	new	old	mid	new	old	mid	new
poor	old	1	0	0	0	0	0	0	0	0	0	0	0
poor	mid	0.5	0.5	0	0	0	0	0	0	0	0	0	0
poor	new	0	0.7	0.3	0	0	0	0	0	0	0	0	0
pass	old	0.3	0	0	0.7	0	0	0	0	0	0	0	0
pass	mid	0	0	0	0.5	0.5	0	0	0	0	0	0	0
pass	new	0	0	0	0	0.7	0.3	0	0	0	0	0	0
good	old	0	0	0	0.3	0	0	0.7	0	0	0	0	0
good	mid	0	0	0	0	0	0	0.5	0.5	0	0	0	0
good	new	0	0	0	0	0	0	0	0.7	0.3	0	0	0
excel	old	0	0	0	0	0	0	0.3	0	0	0.7	0	0
excel	mid	0	0	0	0	0	0	0	0	0	0.5	0.5	0
excel	new	0	0	0	0	0	0	0	0	0	0	0.7	0.3

Intuition for $Pr(S_t | S_{t-1}, A=None)$

- When nothing is done, we want to model that the component generally stays the same, but gets “older” as time passes
- To model this properly, we have to decide what a time slice represents
- In a factory setting, let’s assume we do machine inspections once every 4 months or so. In this case, a time step is about 4 months.
- Here, we can model that components will generally get slightly older and performance degrading. But good quality products may also keep its performance. So we have some uncertainty in the model for this reason

Reward(S,A), in range [0,10]

When A = Replace

S		R
poor	old	0
poor	mid	0
poor	new	0
pass	old	2
pass	mid	2
pass	new	2
good	old	4
good	mid	4
good	new	4
excel	old	6
excel	mid	6
excel	new	6

When A = Fix

S		R
poor	old	2
poor	mid	2
poor	new	2
pass	old	4
pass	mid	4
pass	new	4
good	old	6
good	mid	6
good	new	6
excel	old	8
excel	mid	8
excel	new	8

When A = None

S		R
poor	old	4
poor	mid	4
poor	new	4
pass	old	6
pass	mid	6
pass	new	6
good	old	8
good	mid	8
good	new	8
excel	old	10
excel	mid	10
excel	new	10

Intuition for R(S,A)

- Generally, higher performance will yield higher rewards
  - Excellent: 10
  - Good: 8
  - Pass: 6
  - Poor: 4
- Remember to incorporate a cost to replacing (high cost, perhaps -4) and fixing (low cost, -2)
- Reward should have nothing to do with the age of the component

- Note: you can enumerate all the states/actions and assign the rewards, or you can write if-statements to define R (just make sure you don’t forget any state/action)