Week 9 Lab

Bookmark this page

Week 9 Lab

(1.0 points possible)

State Machines and MDPs

For this lab:

- You will need to understand the material in the notes on state machines and Markov decision processes.
- We suggest that you write down your answers/explanations and make sure you discuss and understand them during the Lab Check-off.

1) MDP Formulation

We will try to model some aspects of a very simple factory as a Markov decision process.

There is a single machine that has three possible operations: "wash", "paint", and "eject", with corresponding buttons. Parts are put into the machine, and each time you push a button, something is done to the part. It's an old machine, and not very reliable. However, it has a camera inside that can clearly detect what is going on with the part and will output the state of the part: either dirty, clean, or painted.

In this question, you will devise a policy that will take as input the state of a part and select a button to press, until finally you press the eject button (which sends you to a state which always transitions to itself and gets zero rewards).

- All parts start out dirty.
- If you perform the "wash" operation on any part, whether it's dirty, clean, or painted, it will end up clean with probability 0.9 and otherwise become dirty.
- If you perform the "paint" operation on a clean part, then with probability 0.8 it becomes nicely painted, with probability 0.1 the paint misses but it stays clean, and with probability 0.1 it dumps rusty dust all over the part and it becomes dirty.
- If you perform the "paint" operation on a painted part, it stays painted with probability 1.0.
- If you perform the "paint" operation on a dirty part, it stays dirty with probability 1.0.
- If you perform an "eject" operation on any part, the part comes out of the machine and this fun game is over.

You get reward +10 for ejecting a painted object, reward 0 for ejecting a non-painted object, and reward -3 for every action that is not "eject".

1A) Write out a careful and complete specification of the **state space**, **action space**, **transition model**, and **reward function**. Provide both a state diagram and a transition matrix.

1B) Use your intuition to find the infinite horizon optimal policy when the discount factor $\gamma = 1$ (no discount). What is its format?

1C) How does the policy from 2 change if $\gamma = 0.1$?

1D) How does the policy from 2 change if the horizon is 2?

Check this box and submit when you have finished question 1.

Ask for Help 100.00%

You have infinitely many submissions remaining.

Solution:

Check this box and submit when you have finished question 1.

Explanation:

You should be ready to discuss your answers to this question during your checkoff.

2) Simple Value Iteration Example

We are going to look at a couple of two-dimensional "grid-world" examples, in which there is a robot that can move North, South, East, or West (N, S, E, W). It cannot move off the board. The transitions are somewhat noisy; when commanding a move there is a small chance that you will end up in one of the neighbor states of the desired state.

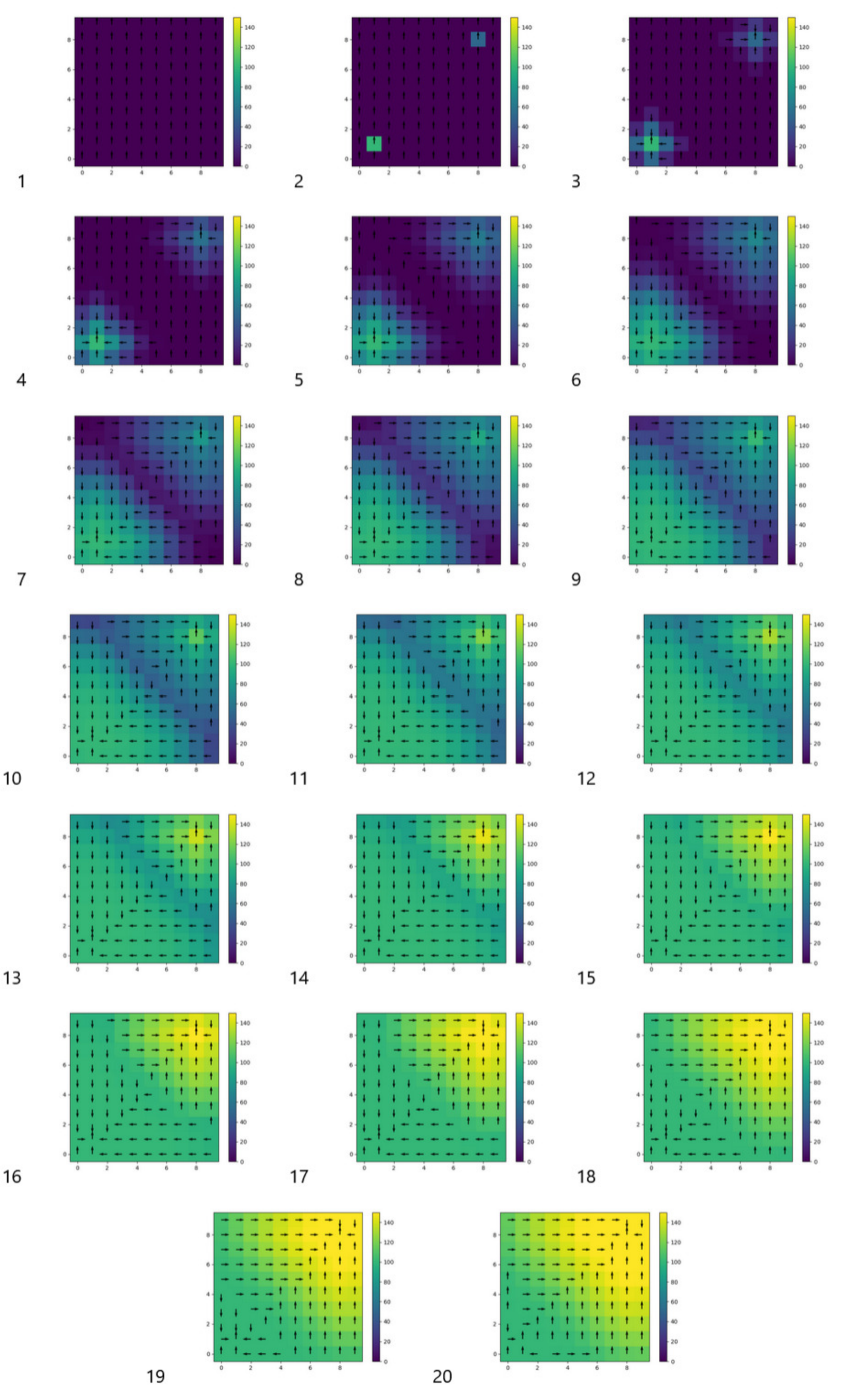
We have defined a special subclass of `MDP` called `GridWorld` for representing these MDPs. One can specify a particular "floor plan" with a list of strings of characters. Consider this world:

```
['..........',
 '........*.',
 '..........',
 '..........',
 '..........',
 '..........',
 '..........',
 '..........',
 '.$........',
 '..........']
```

Each character corresponds to a square in the grid. The meanings of the characters are:

- '.' : a normally habitable square, from which the robot can move.
- '$' : reward of 100; a terminal state, every action leads to a zero-reward state that cannot be escaped.
- '*' : reward of 50 and next state is chosen uniformly at random from all occupiable states (this reward can be claimed multiple times).

These are plots of the values of the states as we run 20 iterations of value iteration with $\gamma = 1$. The arrows represent the current best policy, pointing N, S, E, or W. Be sure you understand how the policy learned on iteration i relates to the value function for horizon i.



2A) In the first picture all the values are zero. After one iteration we have two non-zero states. What are they? What values do they have?

2B) What happens in roughly iterations 2 - 7?

2C) Look at iteration 11. Why does the upper right state now have higher value than the lower left one?

2D) What's happening around iteration 16?

2E) What's happening around iteration 20?

2F) If we keep going, what will the final map look like? Do we expect the robot to eventually terminate?

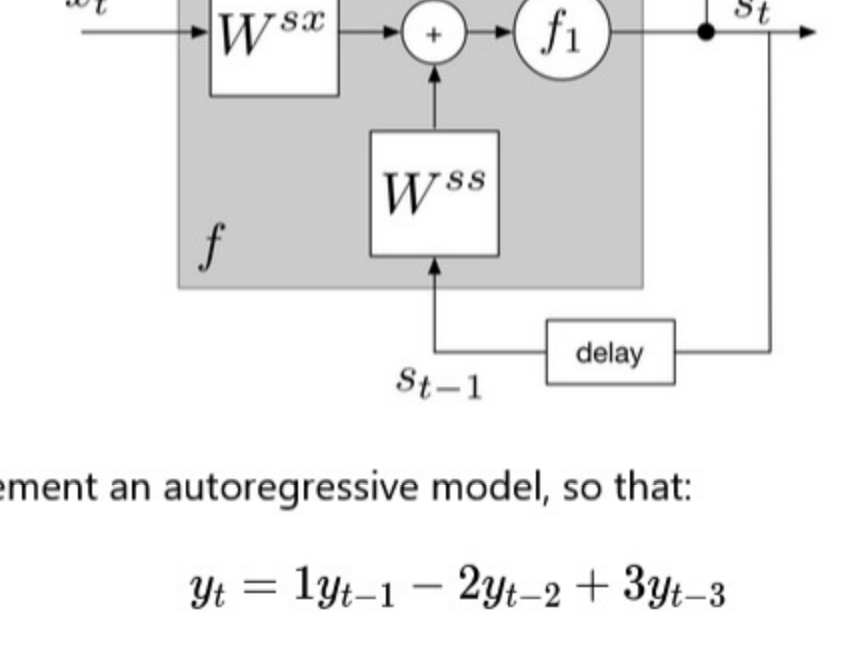
3) Autoregressive RNN

A neural network is a (learned) function that maps from input vectors to output vectors. A recurrent neural network (RNN) is a (learned) state machine that maps input sequences to output sequences. We will learn more about RNNs in week 11; for this lab, we will look at a very simple instance, relating to MDPs. In particular, an RNN has a transition function and an output function, each of which is defined in terms of weight matrices, offset vectors and activation functions, analogously to standard neural networks.

The behavior is defined as follows:

$$s_t = f_1(W^{ss}s_{t-1} + W^{sx}x_t + W_0^{ss})$$
$$y_t = f_2(W^o s_t + W_0^o)$$

where f_1 and f_2 are two activation functions, such as linear, softmax, or tanh. This arrangement is shown below, illustrating that it is an instance of the state machine model that we saw in the Week 9 exercises.



We would like to use an RNN to implement an autoregressive model, so that:

$$y_t = 1y_{t-1} - 2y_{t-2} + 3y_{t-3}$$

We will use an RNN in which:

$$x_t = y_{t-1}$$

This is what makes it "autoregressive." Assume that x_t is a scalar (1x1).

3A) If $y_0 = 5$ and all previous y values are 0, what are y_1, y_2, y_3?

Enter a list of three numbers for $[y_1, y_2, y_3]$: [5, -5, 0]

Submit View Answer Ask for Help 100.00%

You have infinitely many submissions remaining.

3B) What is the best choice for f_1, where f_1 is one of the common activation functions (linear, softmax, tanh, sigmoid, relu) we've studied? Be prepared to explain your answer.

linear

Submit View Answer Ask for Help 100.00%

You have infinitely many submissions remaining.

3C) What is the best choice for f_2? Be prepared to explain your answer.

linear

Submit View Answer Ask for Help 100.00%

You have infinitely many submissions remaining.

3D) What is the smallest dimensionality for the state s that will allow this to be implemented exactly?

Enter a list of two numbers [a, b] for the dimensions $a \times b$ of state s :

[3, 1]

Ask for Help 100.00%

You have infinitely many submissions remaining.

Solution: `[3, 1]`

Explanation:

Be prepared to discuss your answer during the checkout.

3E) Provide matrices W^{ss}, W^{sx}, W^o, W_0^{ss} and W_0^o that implement this model. You should also specify the initial state vector s_0 consistent with $y_0 = 5$ for your model; your initial state vector may contain non-zero elements.