Assignment 2 Part 3

Machine, Data and Learning

Tejasvi Chebrolu, Ruthvik Kodati

2019114005, 2019101035

Making the A matrix:

Procedure -

- We start by making a matrix of dimension 1936 x 600, where 1936 is the number of state-action pairs and 600 is the number of states.
- The matrix's structure consists of a row for each of the state-action pairs. In each of these rows, for states that are not end states, the value of the next state's index will be the probability of reaching that state. However, the value of the current state index will be the probability of going to a different state. For end states, the value of the current state index will be 1. In both cases, the rest will be 0. This ensures that the flow from the end state is 1 and the flow from all other states is 0.
- To construct the A matrix, we iterate over all the possible states of our system and for each state, we add a vector of size (total possible state-action pairs) with appropriate values. The values of vector are initialized to zero.

Pseudo Code -

```
for state in all states:
  for action in state[actions]:
    vector[state-action index] -= probability of action
    vector[state index] += probability of action
    add vector
```

For actions that result in the same state, the probability is set to 0 to avoid the self-looping transitions. The matrix formed from the procedure above is then transposed to

get the final A matrix.

Finding the policy:

- Initially, we enumerated all of our states from 0 to 600.
- After this, we kept track of the number of state-action pairs from each of the states that we encounter. Also, we store which row corresponds to which action from the state.
- Let's say a state's pairs occupied the i^{th} to j^{th} rows. The resultant X vector from our LP would contain the expected rewards in the corresponding i^{th} to the j^{th} indices. Then, taking the argmax from this range would result in the best action from our current state.
- Finally, we can find the policy by doing this for all the states.

Multiple Policies:

There are numerous reasons why there can be multiple policies.

1. We can alter the alpha matrix to make it start from a different state or even have a probabilistic start. This would change our policy.
2. Changing the definition of argmax to take the first maximum argument instead of the last or vice versa will have an effect on our policy.
3. The order in which we consider our states and actions affects what the final policy is.