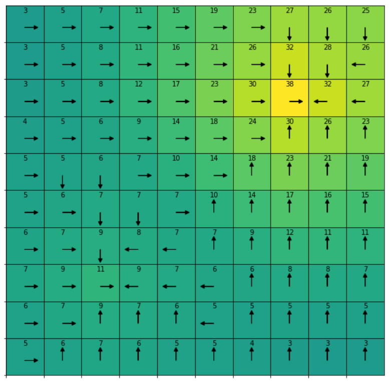
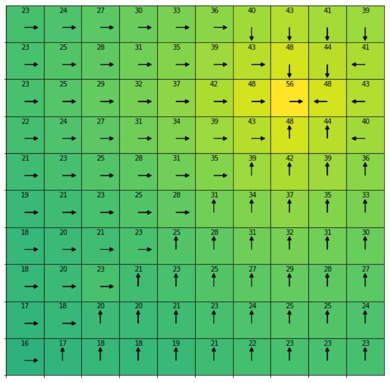
Project 2 - Reinforcement Learning

Ken Oung Yong Quan

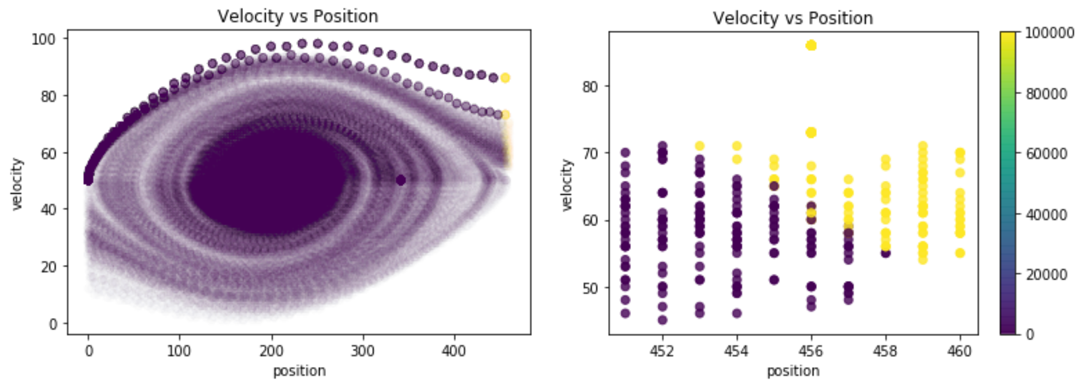
Small - GridWorld [Gauss-Seidel - 0.156s for 50 iterations]
For the GridWorld dataset, we were told that the state transitions were consistent across all different tiles. From the dataset, we can infer that the probability of moving in the direction intended is 0.6, and the remaining 0.4 is split between the remaining possible moves. It is also easy to deduce that the two reward states are State 23 and State 78, with rewards of 3 and 10 respectively.

With the reward and state transition functions, we simply apply Gauss-Seidel value iteration. The figures below are the result after 10 iterations and 50 iterations respectively. The policy had converged by the 50th iteration.





Medium - MovingCar [Value Iteration - 6809 iterations - 47.7s]
In the moving car problem, we can work backwards from the given `1+pos+500*vel` to obtain the discretized velocity and position values. By plotting the values on a scatter plot colored with the reward score, we can identify the bright yellow patch in the corner where we obtain the flag. By observation, we estimate that the flag is obtained when `pos+0.3*vel>475`. For the flag state, we assign a reward of 10000. For all other states, the reward corresponds to the action taken. e.g. action 1 and 7 always results in a fixed reward of -255.



The state transitions are estimated using binned values of position and velocity. Given a (position,velocity,action) triplet, we compute the distribution of change in position and velocity, then use those values to compute the next state.

Using the reward and state transitions, we perform value iteration.

In order to perform the computations in reasonable time, we use a sparse matrix representation for the state transition probability matrix.

Large - Secret [Q-learning - Batch update - 93s]
I performed Q-learning with a learning rate of 0.1.

In order to determine the optimal policy given the learned expected utilities, we assume unseen state action pairs to be zero. If a state is completely unseen, we assign a random action to that state.