**CS7641 Unsupervised Learning HW4**

Aleksandr Plokhikh ([aplokhikh3@gatech.edu](mailto:aplokhikh3@gatech.edu))

**Introduction, dataset selection, data preprocessing**

The goal of the assignment is to study solutions for Markov Decision Problems (MDP), via model based algorithms such as Value Iteration (VI), and Policy Iteration (PI), and model free reinforcement learning algorithm.

As a first problem I took Frozen Lake model, since this model works with multidimensional world (in the general case), and have uncertainty and stochasticity in movements. It lays a fundament to robotics, car autopilots operation, etc. All futurologist of the past (until 2022) thought that the next breakfruth in AI will be in robotics, and there were prerequsits for it like work of Boston dynamic, but great leap came from the completely other thing of AI fild, and robotics is lagging behind probably because to teach it we need a real interaction with the physical world, good exploration-exploatation models and balance, and we can not achive it with extencise calculations on GPUs. I used 8x8 Frozen lake model with 64 states which considered to be a small problem.

The second problem is Black Jack. Of cause the problem have been solved a million times before and on average you can not beat the house, but we probably all saw movie “21” where guys counted cards in Las Vegas and turned result expectancy on their side. Release of othe cards from the deck changes the actual value of the MDP state. We can add realsed cards as a prehistory and make new MDP states based on it. The state space in this case can be as high as trillions instead of the basic 290. MDP with trillions states based on prehistory of released cards can be useful for finding the best strategy but it will be vary hard to find one, so best judjment should be applied, and simple counting like in the movie can be helpful and increases number of states to only several thousands, while providing a better strategies than the default one with no knowledge of prehistory. I do not things that guys in the movie actually solved MDP for most popular “hot” states of the deck so it will be interesting to evaluate overall. Plus basic strategies do not push player to the end of the game faster, and treats the world as we have unlimited time (works with gamma=1) but in the real life aand as in case of movie “21” if you count cards, the casino security may come after you, and you may have only few minutes left to finish several games and escape with prize before got bitten by security team. In thi cases we may want o finish games more quickly and sacrifice the persue of optimal strategy which may require reveal of additional cards. For that we can solve MDP with discounted rewards (gamma<1). So black jack problem is more interesting, undiscovered, and hard than it may seem. In the scope of the assignment I will explore only standard case with 290 states which is already considered big according to asiignment note, and leave much harder cases for further investigation.

Frozen lake value iteration

At condition of no discontinued reward (gamma=1) the state value of the start point is 1 which is equal to the reward of the end which says that if we do not have discontinued reward and allowed to play as long as we want we are guarantied to finish the game at the gall, and will not fall in the hole. Since on the whole map only goal state has non zero immediate reward than positive state values gradually spread across the map with every step of VI, until saturation with logarithm like manner which is seen on the plot of median state value vs number of iterations. Algorithm takes 740ms and 1425 iteration to converge with the standard epsilon of 1e-10. With tightening/loosing epsilon requirents the convergence time will exponentially increase/decrease respectively which clearly visible form the mean state value graph which clearly has logarithmical behavior. No epsilon skew is interesting to make here the behavior is obvious. What interesting to see is when the policy convergence to the optimal one because it is always nice to know the precice expectancy reward at the state but it may not worth extra calculation cost if further improvement of the state value to the true one does not change policy which may be already optimal, and wthis extra calculations in this case does not improve our performance in the game since we use the same policy anyway