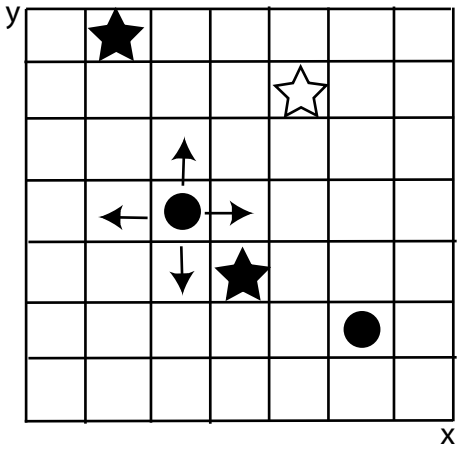
Machine Learning Final

This exam is open book. You may bring in your homework, class notes and textbooks to help you. You will have 1 hour and 15 minutes. Write all answers in the space provided. Please make sure YOUR NAME is on anything you turn in. Square brackets [] denote the points for a question.

1. **Kohonen Maps**
   (a) [10] In the Kohonen Map algorithm, a set of points is mapped onto data points $\mathbf{x} = \{x_1, \ldots, x_N\}$ using a topology that has a certain dimension. For example in the traveling salesman problem the dimension is one, but in other problems the dimension is two or greater. Write the outline of a Kohonen Map algorithm that takes the dimension of the topology as a parameter.
   (b) [5] What is the maximum dimension of the topology that is reasonable?
   (c) [10] Suppose the topology should not be uniform for all of the data set, but in some regions should be two-dimensions and in other regions three dimensions. How could you modify the basic algorithm to adjust these dimensions on line?

2. **Reinforcement Learning**

In a simple 2D grid world, robots(circles) get rewarded for collecting black stars and punished for collecting white stars.



The robots are use standard reinforcement learning such that each robot k can be described by a Markov Decision Process $MDP = \{S_k, A_k, T_k, R_k\}$ where S_k is the state space, A_k is the action space, T_k is the transition function and R_k is the reward function.

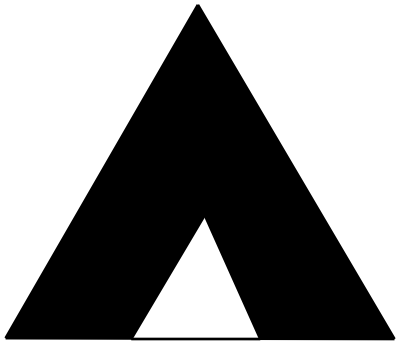
(a) [10] Specify a possible S_k, A_k, T_k and R_k for these robots.

(b) [10] If two robots are to be *considered as a single robot*, show formally how their information can be combined. What is the new $MDP = \{S, A, T, R\}$ in terms of the old?

(c) [5] Now suppose the 'merged' robot given by your $\{S, A, T, R\}$ is to be split up into two robots. Is there a problem here? Say why or why not.

3. **Genetic Algorithms**

In the diagram below is an abstract representation of a program where white denotes code that is never executed.



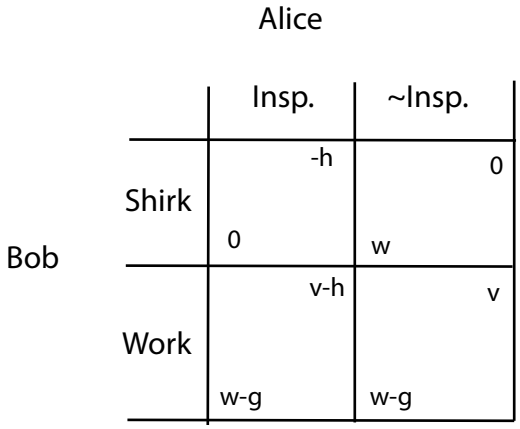
(a) [10] In a genetic algorithm a small set of examples or *fitness cases* are used to evaluate an individual. Why not just use all the possible inputs?

(b) [5] Why is it useful to have fitness cases evolve?

(c) [10] In genetic programming, lots of an individual program is never executed. Should this code be pruned? Give a reason why it might be a good idea and a reason it might not be.

4. **Games**

The Work or Shirk game is given by:



where the worker chooses to Shirk with probability x and the inspector chooses Inspect with probability y.

(a) [10] Derive two formulas that reflects the payoffs expected by the Worker and Inspector.

(b) [15] Use calculus to derive the best probabilities for both players.