Homework 3: Machine Learning 2D5362

Handed out: Tuesday, 5.12.00
Due: Tuesday, 12.12.00 : 13:30
Name:

1. Implement a genetic algorithm that solves the brachystrochrone problem, namely finding the optimal curve between two points $(x_0,y_0)$ and $(x_n,y_n)$ which a point mass under the force of gravity travels in minimal time. Instead of implementing the entire GA code yourself I recommend to use the MIT GaLib.
It should compile without problems under SUN OS 5.3 and Linux. Read the documentation file galibdoc.pdf that comes with it. The genome class **GABin2DecGenome** and for the GA itself **GASimpleGA**, in which case you merely have to design the fitness function. If $(x_i,y_i)$ and $(x_{i+1},y_{i+1})$ are the start and end point of and $v_i$ is the velocity of the mass at point $(x_i,y_i)$ then due to the conservation of energy the new velocity $v_{i+1}$ at point $(x_{i+1},y_{i+1})$ becomes $v_{i+1}=v_i$ ? $(2\ g\ |(y_i-y_{i+1})|)^{1/2}$. If we further assume that the two points are connected by a straight time we can compute the travel time for the i-th segment as

$$t_i=((x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2)^{1/2}/((v_i+v_{i+1})/2).$$

Assume that $v_0=0$ and $y_n<y_0$. Use equidistant spacing along the x-axis between the track points and let the GA only optimise the heights of the inner track points $\{y_1,\ldots,y_{n-1}\}$.

2. Implement a genetic algorithm that evolves a game playing strategy for the prisoner's dilemma game. Evaluate each genotype by letting it play a game of k repeated rounds against each of the other members in the current population. Encode a strategy that makes a decision to cooperate or defect based on the outcome of the previous round played against the same opponent. Use the following pay-off matrix cooperate/cooperate=3, cooperate/defect=1, defect/cooperate=5, defect/defect=2, for own action / opponent's action. I recommend using the class **GA1DBinaryStringGenome** to encode the strategy.

3. Assume the following grid world of a 4x4 square. In each cell the agent can choose one of the four possible actions (North, West, South, East) to move to a neighboring cell. If the agent tries to move beyond the boundaries it remains in the original but incures a penalty of –1. There are two special cells A and B from which the agent is "beamed" to A' and B' no matter which action it chooses. For this transit it receives a reward of +10 (A to A') and +5 (B to B'). For the regular moves the reward is zero. The problem has an infinite horizon, there are no terminal states and the agent continues forever. Assume a discount factor of ?=0.9.
   a. Compute the value function $V^?$ for an equi-probable policy in which all actions in all states have the same constant probability ?(s,a)=1/4.
   b. Compute the optimal value function $V^?$ and policy ?* using value or policy iteration.
   c. Compute the optimal value function $V^?$ and policy ?* using value or policy iteration, but for a non-deterministic state transition function. Assume that with probability p=0.7, the agent moves to the "correct" square, but with probability 1-p=0.3 it is pushed to a random neighboring square.