CSCE 5043, Advanced Artificial Intelligence, Fall 2017, Midterm Exam

1. (5 points) What is your name?

$$Q(i,a) \leftarrow (1-\alpha^k)Q(i,a) + \alpha^k \left[ r(i,a,j) + \gamma \max_{b \in A(j)} Q(j,b) \right]$$

2. (5 points) Note the Q-update equation above. Consider a game in which an agent must choose between seeking a small immediate reward or seeking a large reward in the distant future. How would you adjust the parameters of the Q-learning update equation (above), to cause the agent to seek the larger reward? (Circle one.)

- Use a value for $\alpha$ (alpha) close to 0.
- Use a value for $\alpha$ (alpha) close to 1.
- Use a value for $\gamma$ (gamma) close to 0.
- Use a value for $\gamma$ (gamma) close to 1.
- Use a value for *k* close to 0.
- Use a value for *k* close to 1.
- Use a value for Q close to 0.
- Use a value for Q close to 1.
- Use a value for *A*(*j*) close to 0.
- Use a value for *A*(*j*) close to 1.
- The parameters do not matter. The "max" term causes it to seek the larger reward.

3. (5 points) How would you make a Q-learner robust to non-determinism? (For example, suppose the action it actually performs is sometimes different from the one it chooses. Circle one answer.)

- Use a value for $\alpha$ (alpha) close to 0.
- Use a value for $\alpha$ (alpha) close to 1.
- Use a value for $\gamma$ (gamma) close to 0.
- Use a value for $\gamma$ (gamma) close to 1.
- Use a value for *k* close to 0.
- Use a value for *k* close to 1.
- Use a value for Q close to 0.
- Use a value for Q close to 1.
- Use a value for *A*(*j*) close to 0.
- Use a value for *A*(*j*) close to 1.
- Use rewards instead of penalties to reinforce robust behavior.

4. (15 points) Draw a diagram of an actor-critic model suitable for doing reinforcement learning with continuous actions. (There are multiple correct ways to do this.) Use boxes to indicate function-approximating learning models. Identify your models. Use arrows to indicate the flow of information. Identify the inputs and outputs of each model. Also, state how the target output values for each model in your diagram may be computed.

5. (15 points) Which conditions are necessary for Q-learner with an epsilon-greedy exploration strategy to be able to guarantee to find the optimal policy? (Circle all that are necessary. Don't circle any that are not necessary.)

- Always exploit. Never explore.
- Every state must be reachable with some non-zero probability.
- Actions must be continuous.
- Actions must be multi-dimensional.
- The Q table must use sufficient granularity to capture the state space.
- Training must continue until convergence.
- An approximately infinite amount of memory is required.
- A neural network (or some other continuous model) must be used to implement the Q-table.
- There must not be any negative rewards.
- The learning rate (alpha) must be sufficiently close to zero if there is any non-determinism in the actions.
- There must be no negative rewards.
- An actor-critic model must be used.
- A sufficiently large experience-replay buffer must be used.
- Values drawn from a spherical distribution must be used to initialize the Q table.
- The discount factor must be set to exactly the same value used in the definition of "optimal" for the policy.
- The problem must be one that humans are capable of solving.
- The set of possible actions must be the same in every state.
- Rewards must not be a function of the action selected.
- It must be run on a super-computer.

6. (5 points) Which type of learning is performed by Principal Component Analysis?

- supervised learning
- unsupervised learning
- reinforcement learning

7. (5 points) Which type of learning is performed by Q-learning?

- supervised learning
- unsupervised learning
- reinforcement learning

8. (5 points) Which type of learning is performed by *k*-NN?

- supervised learning
- unsupervised learning
- reinforcement learning

9. (5 points) Is *k*-NN a universal function approximator? That is, can it approximate any function with arbitrary precision?

- No, some functions may exist that are too complex for *k*-NN to approximate.
- Yes, but only if a neighbor-finding acceleration structure, such as a kd tree, is used.
- Yes, but only if the value for "k" is sufficiently large.
- Yes, but only if enough training data is provided.
- Yes, but only if the right distance metric is selected.

10. (5 points) Which of the following adjustments to a kd tree could make it fail to find the correct k-nearest neighbors? (Circle all that could make it give incorrect results.)

- Use random divisions of the data when building the kd tree.
- When searching for neighbors, continue visiting all of the kd nodes until they have all been visited.
- When searching for neighbors, continue visiting all of the kd nodes in the kd tree until they have all been visited, and visit them in a different order.
- Use a distance metric that does not satisfy the triangle inequality.
- Stop as soon as k+1 points have been evaluated.
- Each time you test a point, also test another point randomly drawn from the data.

11. (5 points) Given that,

$$x^0 + x^1 + x^2 + x^3 + \ldots = 1 / (1 - x)$$

if some Q-learner uses an exploration rate of 0.05, a learning rate of 0.2, and a discount factor of 0.9, and receives a reward in the range [-3.14, 3.14] at every time step, what is the theoretical maximum value (discounted expected horizon value according to the Bellman equation) that may be assigned to any state? (Knowing this upper bound could be useful, for example, if you need to normalize your Q-values to make the suitable for use with a neural network.)

12. (10 points) The Euclidean distance between the two points <1, 2> and <5, 5> is 5. If we adjust these points minimally and equally to make the Euclidean distance between them be 1, what are their new values?

13. (5 points) The $L^1$ distance (a.k.a. Manhattan distance or taxicab distance) between <1, 2> and <5, 5> is 7. Manhattan distance is useful for computing taxi fares in cities arranged in rectangular blocks. When else might Manhattan distance be a better choice than Euclidean distance? (Circle the best one.)

- For classification in high dimensional spaces.
- For classification in low dimensional spaces.
- For regression with missing values.
- For regression in nondeterministic action spaces.
- For unsupervised learning with categorical values.
- When rotational invariance is important.
- When computing an unbiased estimator of covariance.

14. (5 points) Which of the following statements accurately characterize manifold learning? (Circle all that are accurate.

- After manifold learning, data points should be arranged approximately into a low-dimensional rectangle.
- After manifold learning, distances between points in local neighborhoods will be approximately the same as before.
- Manifold learning attempts to preserve Euclidean distance between non-neighboring points.
- Manifold learning is always done as an iterative relaxation process.
- If the intrinsic dimensionality is as large as the extrinsic dimensionality, then manifold learning has no value.

15. (5 points) What is the value of adding $L^1$ regularization to an associative memory model?

- It causes the model to approximate the natural distribution that occurs in the data.
- It helps the model fit to the data better.
- It promotes sparse encodings.
- It reduces the “energy” of the system.
- It lets you clamp known values.