Problem Set 6

CS 6375

Due: 5/9/2020 by 11:59pm

Note: all answers should be accompanied by explanations for full credit. Late homeworks will not be accepted.

Problem 1: Markov Decision Processes (30pts)

1. Consider a Markov decision process with four states s_1, s_2, s_3, s_4 and four actions a_1, a_2, a_3, a_4 with the following transition and reward functions.

   - $T(s_i, a_j) = s_j$ for all $i, j \in \{1, 2, 3, 4\}$.
   - $R(s_1, a_2) = R(s_2, a_3) = 1$, $R(s_3, a_1) = R(s_4, a_1) = -1$, the rewards of the form $R(s_i, a_i) = 0$ for all states $i \in \{1, 2, 3, 4\}$, and the rewards for the remaining transitions are equal to .5.

   Using the above MDP, answer the following questions.

   (a) How many possible deterministic policies are there for this MDP?

   (b) For $\gamma = .8$, find the optimal value function V^* and an optimal policy π^*. Is there a unique optimal deterministic policy?

   (c) How does the optimal policy change if $\gamma = .01$?

2. For any MDP and any two policies π_1 and π_2, show how to construct a policy π_3 such that $V^{\pi_3}(s) \geq V^{\pi_1}(s)$ for all $s \in S$ and $V^{\pi_3}(s) \geq V^{\pi_2}(s)$ for all $s \in S$.

3. As we saw in class, Markov decision processes make decisions based on the current state of the environment and a chosen policy. Suppose that you are given an MDP, but you would like the agent's decision to depend on the last two states of the environment instead of just the last. Can this requirement be formulated in the MDP framework? Explain why or why not.

Problem 2: Gaussian Mixtures vs. k-means (50pts)

For this problem, you will use the `leaf.data` file provided with this problem set. This data set was generated from the UCI Leaf Data Set (follow the link for information about the format of the data). The class labels are still in the data set and should be used for evaluation only (i.e., don't use them in the clustering procedure), but the specimen number has been removed. You should preprocess the data so that the non-label attributes have mean zero and variance one.

1. Train a k-means classifier for each $k \in \{12, 18, 24, 36, 42\}$ starting from twenty different random initializations (sample uniformly from $[-3, 3]$ for each attribute) for each k. Report the mean and variance of the value of the k-means objective obtained for each k.

2. Train a Gaussian mixture model for each $k \in \{12, 18, 24, 36, 42\}$ starting from twenty different random initializations (random mean and covariance matrix equal to the identity matrix) for each k. Report the mean and variance of the converged log-likelihood for each k.

3. Looking at the true labels, for $k = 36$, which of these two models might you prefer for this data set?

4. Random initializations can easily get stuck in suboptimal clusterings. An improvement of the k-means algorithm, known as k-means++, instead chooses an initialization as follows:

   (a) Choose a data point uniformly at random to be the first center.

   (b) Repeat the following until k centers have been selected:

      i. For each data point x compute the distance between x and the nearest cluster center in the current set of centers. Denote this distance as d_x.

      ii. Sample a training data point at random from the distribution p such that $p(x) \propto d_x^2$. Add the sampled point to the current set of centers.

   Repeat the first two experiments using this initialization to pick the initial cluster centers for k-means and the initial cluster means for the Gaussian mixture model. Does this procedure result in an improvement in either case?

5. Suppose that, instead of allowing the covariance matrix of each mixture component to be an arbitrary positive definite matrix, we require each covariance matrix to be a diagonal matrix such that all of its diagonal entries are strictly larger than zero. Explain how to modify the EM algorithm and derive the new updates for this special case.

Problem 3: Neural Networks (20pts)

1. Using only perceptrons, construct a neural network for the following input/output pair: Takes 10 binary inputs with one binary output which is 1 if the number of ones in the input is divisible by four and zero otherwise.

2. Suppose that you add an ℓ_2 regularizer, i.e., $\frac{\lambda}{2}||w||_2^2$, to the squared loss minimization problem for fitting neural networks in class. How does the backpropagation algorithm change?

3. Given a binary classification task, explain the difference between fitting a neural network with a single perceptron hidden unit to data versus using the perceptron algorithm to fit a linear classifier to the same data.

Bonus (15pts): Generate a data set for this problem and use MATLAB or Python (you can use built-in functions for backpropagation) to learn the weights of a sigmoid or relu (e.g., $\max\{0, \cdot\}$ or its smooth analog) neural network with the same structure as you produced for the first part of this question. You should use a cross entropy loss. How do the learned weights compare to your perceptron solution as you vary the size of the training set?

Course Evaluation:

If you haven't done so already, please go to `eval.utdallas.edu` and provide feedback on the course.