AI (Fall 2018) - Assignment 4
Machine learnning
Due: Due: 11:59pm, Sunday, 23, Dec, 2018

1. A TV channel collected data as the following table from a viewer to investigate the preference. Use DECISION-TREE-LEARNING algorithm to learn a decision tree that optimizes the information gain at each split.

   (a) Chenyang correctly chose the first attribute **Lawyers**, but he didn't get the points because he left out the information gain at this step. Compute the information gain.

   (b) Finish the entire tree and show the computations.

Example	**Comedy**	**Doctors**	**Lawyers**	**Guns**	**Likes**
e1	false	true	false	false	false
e2	true	false	true	false	true
e3	false	false	true	true	true
e4	false	false	true	false	false
e5	false	false	false	true	false
e6	true	false	false	true	false
e7	true	false	false	false	true
e8	false	true	true	true	true
e9	false	true	true	false	false
e10	true	true	true	false	true
e11	true	true	false	true	false
e12	false	false	false	false	false

2. Consider the candy example from the lecture. Assume that the prior distribution over $h_1, ..., h_5$ is given by $\langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle$. Chenyang had sampled the first 5 candies, and he correctly concluded that the h_{MAP} was h_3. Unfortunately he spilled coffee and the third data point was missing: *lime, cherry, ____, lime, lime.*

   (a) Recover the missing flavor by computation.

   (b) Make predictions for the 6th candy using Bayesian and ML learning, respectively. Show the computations done to make the predictions.

3. Give an algorithm for EM for unsupervised learning that does not store an A array(Figure 1), but rather recomputes the appropriate value for the M step. Each iteration should only involve one sweep through the data set. (Hint: For each tuple in the data set, update all of the relevant $M_i - values$.)

procedure $EM(X, D, k)$
  Inputs
    X set of features $X = \{X_1, \ldots, X_n\}$
    D data set on features $\{X_1, \ldots, X_n\}$
    k number of classes
  Output
    $P(C), P(X_i|C)$ for each $i \in \{1 : n\}$, where $C = \{1, \ldots, k\}$.
  Local
    real array $A[X_1, \ldots, X_n, C]$
    real array $P[C]$
    real arrays $M_i[X_i, C]$ for each $i \in \{1 : n\}$
    real arrays $P_i[X_i, C]$ for each $i \in \{1 : n\}$
  $s :=$ number of tuples in D
  Assign $P[C], P_i[X_i, C]$ arbitrarily
  repeat
                                                        ▷ *E* Step
    for each assignment $\langle X_1 = v_1, \ldots, X_n = v_n \rangle \in D$ **do**
      let $m \leftarrow |\langle X_1 = v_1, \ldots, X_n = v_n \rangle \in D|$
      for each $c \in \{1 : k\}$ **do**
        $A[v_1, \ldots, v_n, c] \leftarrow m \times P(C = c | X_1 = v_1, \ldots, X_n = v_n)$
                                                        ▷ *M* Step
    for each $i \in \{1 : n\}$ **do**
      $M_i[X_i, C] = \sum_{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n} A[X_1, \ldots, X_n, C]$
      $P_i[X_i, C] = \dfrac{M_i[X_i, C]}{\sum_C M_i[X_i, C]}$
    $P[C] = \sum_{X_1, \ldots, X_n} A[X_1, \ldots, X_n, C] / s$
  until termination

Figure 1: Figure 1: EM for unsupervised learning

4. Consider four different ways to derive the value of α_k from k in $Q-learning$ (note that for $Q-learning$ with varying α_k there must be a different count k for each state-action pair).

   - let $\alpha_k = 1/k$
   - let $\alpha_k = 10/(9 + k)$
   - let $\alpha_k = 0.1$
   - Let $\alpha_k = 0.1$ for the first 10000 steps, $\alpha_k = 0.01$ for the next 10000 steps, $\alpha_k = 0.001$ for the next 10000 steps, $\alpha_k = 0.0001$ for the next 10000 steps and so on.

   (a) Which of these will converge to the true Q-value in theory?

(b) Which converges to the true $Q - value$ in practice (i.e., in a reasonable number of steps)? Try it for more than one domain.

(c) Which can adapt when the environment adapts slowly?