Solutions

10-301/601 Machine Learning **Name:**
Summer 2023 **AndrewID:**
Final Practice Problems
August 2, 2023
Time Limit: N/A

Instructions:

- Fill in your name and Andrew ID above. Be sure to write neatly, or you may not receive credit for your exam.
- Clearly mark your answers in the allocated space **on the front of each page.** If needed, use the back of a page for scratch space, but you will not get credit for anything written on the back of a page. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- No electronic devices may be used during the exam.
- Please write all answers in pen.
- You have N/A to complete the exam. Good luck!

Instructions for Specific Problem Types

For "Select One" questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

● Henry Chai

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

● Henry Chai

○ Marie Curie

~~●~~ Noam Chomsky

For "Select all that apply" questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

~~■~~ I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601		10-~~7~~601

1 Learning Theory

1. **True and Sample Errors:** Consider a classification problem with distribution D and target function $c^* : \mathcal{R}^d \mapsto \pm 1$. For any sample S drawn from D, answer whether the following statements are true or false, along with a brief explanation.

   1. **True or False:** For a given hypothesis space $\mathcal{H}$, it is always possible to define a sufficient number of examples in S such that the true error is within a margin of ϵ of the sample error for all hypotheses $h \in H$ with a given probability.

      False. If $VC(\mathcal{H}) = \infty$, then there is no (finite) number of examples sufficient to satisfy the PAC bound.

   2. **True or False:** The true error of any hypothesis h is an upper bound on its training error on the sample S.

      False. We said true error is close to training error, but it might be smaller than training error, so it is not an upper bound.

2. Let X be the feature space and D be a distribution over X. We have a training data set

   $$\mathcal{D} = \{(x_1, c^\star(x_1)), \cdots, ((x_N, c^\star(x_N)))\},$$

   x_i i.i.d from D. We assume labels $c^\star(x_i) \in \{-1, 1\}$.

   Let $\mathcal{H}$ be a hypothesis class and let $h \in \mathcal{H}$ be a hypothesis. In this question we restrict ourselves to $\mathcal{H}$. We use

   $$err_S(h) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(h(x_i) \neq c^\star(x_i))$$

   to denote the training error and

   $$err_D(h) = P_{x \sim D}(h(x) \neq c^\star(x))$$

   to denote the true error. Recall that if the concept class is finite, in the realizable case

   $$m \geq \frac{1}{\epsilon}\left[\ln(|\mathcal{H}|) + \ln\left(\frac{1}{\delta}\right)\right]$$

   labeled examples are sufficient so that with probability at least $1 - \delta$, all $h \in \mathcal{H}$ with $err_D(h) \geq \epsilon$ have $err_S(h) > 0$; in the agnostic case,

   $$m \geq \frac{1}{2\epsilon^2}\left[\ln(|\mathcal{H}|) + \ln\left(\frac{2}{\delta}\right)\right]$$

labeled examples are sufficient such that with probability at least $1-\delta$, all $h \in \mathcal{H}$ have $|err_D(h) - err_S(h)| < \epsilon$.

1. Briefly describe the difference between the realizable case and agnostic case.

   Realizable- the true classifier c^* is in $\mathcal{H}$.

   Agnostic- we don't know whether c^* is in $\mathcal{H}$. It may or may not be.

2. What is the full name of PAC learning? How do ϵ and δ tie into the name?

   "Probably approximately correct." The hypotheses we find with m examples are *probably* (with probability $p \geq 1-\delta$) *approximately* correct, with $err_D(h) \leq \epsilon$

3. **True or False:** Consider two finite hypothesis sets $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathcal{H}_1 \subset \mathcal{H}_2$. Let $h_1 = \arg\min_{h \in \mathcal{H}_1} err_S(h)$ and $h_2 = \arg\min_{h \in \mathcal{H}_2} err_S(h)$. Because $|\mathcal{H}_2| \geq |\mathcal{H}_1|$, $err_D(h_2) \geq err_D(h_1)$.

   False. Since there are more hypotheses in $\mathcal{H}_2$ there might be one that better fits the data than those in $\mathcal{H}_1$.

3. **Fill in the Blanks:** Complete the following sentence by circling one option in each square (options are separated by "/"s):

   In order to prove that the VC-dimension of a hypothesis set $\mathcal{H}$ is D, you must show that $\mathcal{H}$ [can / cannot] shatter [any set / some set / multiple sets] of D data points and [can / cannot] shatter [any set / some set / multiple sets] of $D+1$ data points.

   In order to prove that the VC-dimension of a hypothesis set $\mathcal{H}$ is D, you must show that $\mathcal{H}$ can shatter some set of D data points and cannot shatter any set of $D+1$ data points.

4. Consider the hypothesis set $\mathcal{H}$ consisting of all positive intervals in $\mathbb{R}$, i.e. all hypotheses of the form $h(x; a, b) = \begin{cases} +1 & \text{if } x \in [a, b] \\ -1 & \text{if } x \notin [a, b] \end{cases}$

   1. **Short Answer:** In 1-2 sentences, briefly justify why the VC dimension of $\mathcal{H}$ is less than 3.

      We only need to show 3 points cannot be shattered. Consider 3 points where the two outer points have label +1 and the middle point has label -1.

   2. **Select one:** What is the VC dimension of $\mathcal{H}$?

      ○ 0

      ○ 1

      ○ 2

      C

   3. **Numerical Answer:** Now, consider hypothesis sets $\mathcal{H}_k$ indexed by k, such that $\mathcal{H}_k$ consists of all hypotheses formed by k **non-overlapping** positive intervals in $\mathbb{R}$. Give an expression for the VC dimension of $\mathcal{H}_k$ in terms of k.

      Hint: Think about how to repeatedly apply the result you found in Part (b).

      $2k$

5. **Select one:** Your friend, who is taking an introductory ML course, is preparing to train a model for binary classification. Having just learned about PAC Learning, she informs you that the model is in the finite, agnostic case.

   Now she wants to know how changing certain values will change the number of labelled training data points required to satisfy the PAC criterion. For each of the following changes, determine whether the sample complexity will increase, decrease, or stay the same.

   1. Using a simpler model (decreasing $|\mathcal{H}|$)

      ◯ Sample complexity will increase

      ◯ Sample complexity will decrease

      ◯ Sample complexity will stay the same

      B

   2. Choosing a new hypothesis set $\mathcal{H}^*$, such that $|\mathcal{H}^*| = |\mathcal{H}|$

      ◯ Sample complexity will increase

      ◯ Sample complexity will decrease

      ◯ Sample complexity will stay the same

      C

   3. Decreasing δ

      ◯ Sample complexity will increase

      ◯ Sample complexity will decrease

      ◯ Sample complexity will stay the same

      A

   4. Decreasing ϵ

      ◯ Sample complexity will increase

      ◯ Sample complexity will decrease

      ◯ Sample complexity will stay the same

      A

6. **Errors, Errors Everywhere,** In this question we will consider the effect of training set size and regularization on the performance of a classifier. We will use err_D to denote the true error rate of the classifier and err_S to denote the error on the training set.

   Please provide a **one line justification** to your answer.

7. **T or F**: PAC learning theory allows us to determine with 100% certainty the maximum possible true error of a classifier. False

8. Which one of the following is a guaranteed consequence of regularization in linear regression? (the error metric is mean square error)

   (a) Training error will increase or remain the same as the non-regularized model

   (b) Training error will decrease or remain the same as the non-regularized model

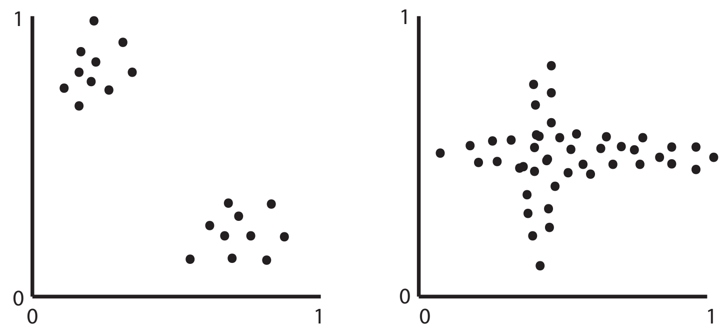
   (c) True error will increase or remain the same as the non-regularized model

   (d) True error will decrease or remain the same as the non-regularized model

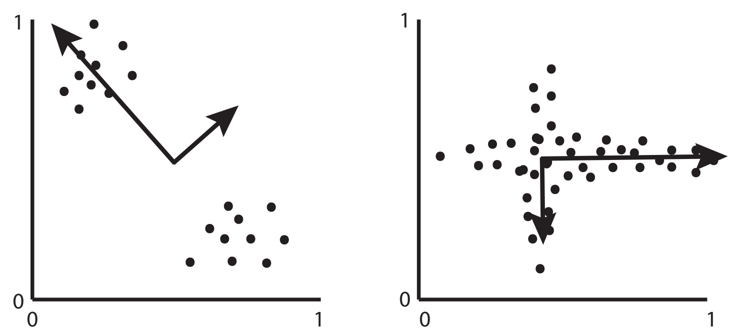
   (a)

2 Principal Component Analysis

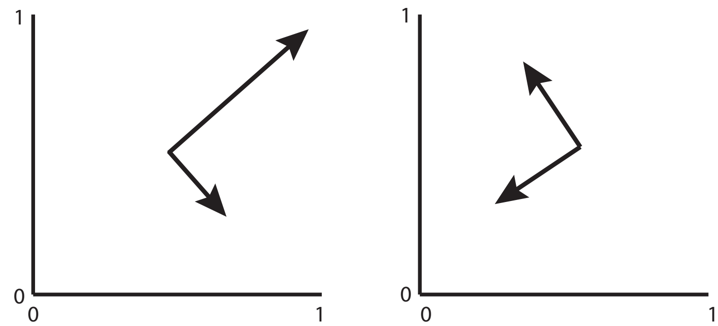
1. 1. Consider the following two plots of data. Draw arrows from the mean of the data to denote the direction and relative magnitudes of the principal components.



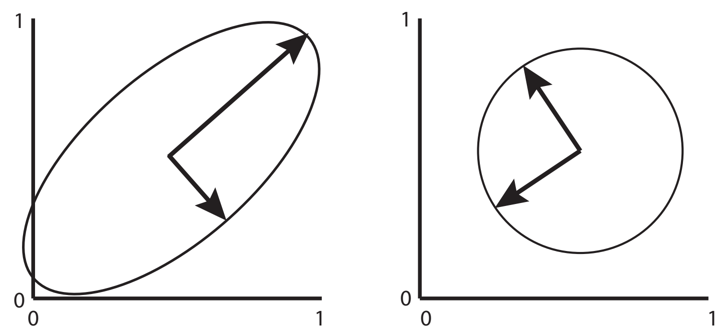
Solution:



2. Now consider the following two plots, where we have drawn only the principal components. Draw the data ellipse or place data points that could yield the given principal components for each plot. Note that for the right hand plot, the principal components are of equal magnitude.



Solution:



2. Circle one answer and explain.

   In the following two questions, assume that using PCA we factorize $X \in \mathbb{R}^{n\times m}$ as $Z^T U \approx X$, for $Z \in \mathbb{R}^{m\times n}$ and $U \in \mathbb{R}^{m\times m}$, where the rows of X contain the data points, the rows of U are the prototypes/principal components, and $Z^T U = \hat{X}$.

   1. Removing the last row of U and Z will still result in an approximation of X, but this will never be a better approximation than $\hat{X}$.

      ◯ True

      ◯ False

      Justify your answer:

      True. As we are removing a principal component of the data when we remove any row from U and Z, we take the variance attributed to that principal component with it. Since variance is always nonnegative, removing some of the variance preseved by a given principal component will increase the reconstruction error of the original data (recall that maximizing the variance preserved is equivalent to minimizing the reconstruction error).

   2. $\hat{X}\hat{X}^T = Z^T Z$.

      ◯ True

      ◯ False

      Justify your answer:

      True. $\hat{X}\hat{X}^T = Z^T U (Z^T U)^T = Z^T U U^T Z = Z^T Z$. Recall that the rows of U are eigenvectors, meaning $U^T U$ has non-zero entries in the main diagonal only. Further, the principal components themselves are unit vectors, meaning the dot product of any eigenvector in U with itself is one. Then the main diagonal of $U^T U$ is all ones. Thus, $U^T U$ is the identity matrix.

3. The goal of PCA is to interpret the underlying structure of the data in terms of the principal components that are best at predicting the output variable.

   ◯ True

   ◯ False

   Justify your answer:

   False. The goal of PCA is to produce an underlyiing structure to the data that preserves the largest amount of variance (or synonymously minimizes the reconstruction error).

4. The output of PCA is a new representation of the data that is always of lower dimensionality than the original feature representation.

   ◯ True

   ◯ False

   Justify your answer:

   False. PCA can produce a representation that is up to the same number of dimensions as the original feature representation.

3 K-Means

1. For **True or False** questions, circle your answer and justify it.

    1. For a particular dataset and a particular k, k-means will always produce the same result, if the initialized centers are the same. Assume there is no tie when assigning the clusters.

        ◯ True

        ◯ False

        Justify your answer:

        True. Every time you are computing the completely same distances, so the result is the same.

    2. k-means can always converge to the global optimum.

        ◯ True

        ◯ False

        Justify your answer:

        False. It depends on the initialization. Random initialization could possibly lead to a local optimum.

    3. k-means is not sensitive to outliers.

        ◯ True

        ◯ False

        Justify your answer:

        False. k-means is quite sensitive to outliers, since it computes the cluster center based on the mean value of all data points in this cluster.

4. k in k-nearest neighbors and k-means have the same meaning.

   ◯ True

   ◯ False

   Justify your answer:

   False. In knn, k is the number of data points we need to look at when classifying a data point. In k-means, k is the number of clusters.

5. What's the biggest difference between k-nearest neighbors and k-means?

   Write your answer in one sentence:

   ---

   knn is a supervised algorithm, while k-means is unsupervised.

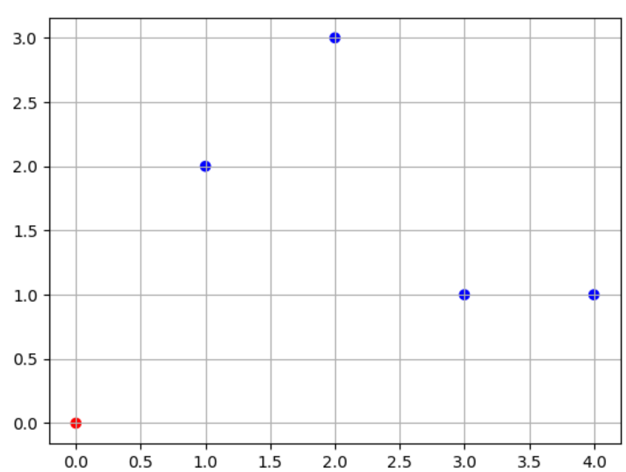
2. In k-means, random initialization could possibly lead to a local optimum with very bad performance. To alleviate this issue, instead of initializing all of the centers completely randomly, we decide to use a smarter initialization method. This leads us to k-means++.

   The only difference between k-means and k-means++ is the initialization strategy, and all of the other parts are the same. The basic idea of k-means++ is that instead of simply choosing the centers to be random points, we sample the initial centers iteratively, each time putting higher probability on points that are far from any existing center. Formally, the algorithm proceeds as follows.

   Given: Data set $x^{(i)}, i = 1, \ldots, N$
   Initialize:
   $\mu^{(1)} \sim \text{Uniform}(\{x^{(i)}\}_{i=1}^N)$
   For $j = 2, \ldots, k$
   Computing probabilities of selecting each point
   $$p_i = \frac{\min_{j'<j} \|\mu^{(j')} - x^{(i)}\|_2^2}{\sum_{i'=1}^N \min_{j'<j} \|\mu^{(j')} - x^{(i')}\|_2^2}$$
   Select next center given the appropriate probabilities
   $$\mu^{(j)} \sim \text{Categorical}(\{x^{(i)}\}_{i=1}^N, \mathbf{p}_{1:N})$$

   Note: n is the number of data points, k is the number of clusters. For cluster 1's center, you just randomly choose one data point. For the following centers, every time you initialize a new center, you will first compute the distance between a data point and the center closest to this data point. After computing the distances for all data points, perform a normalization and you will get the probability. Use this probability to sample for a new center.

   Now assume we have 5 data points (n=5): (0, 0), (1, 2), (2, 3), (3, 1), (4, 1). The number of clusters is 3 ($k = 3$). The center of cluster 1 is randomly chosen as (0, 0). These data points are shown in the figure below.



1. What is the probability of every data point being chosen as the center for cluster 2? (The answer should contain 5 probabilities, each for every data point)

(0, 0): 0
(1, 2): 0.111
(2, 3): 0.289
(3, 1): 0.222
(4, 1): 0.378

2. Which data point is mostly liken chosen as the center for cluster 2?

(4, 1) is mostly likely chosen.

3. Assume the center for cluster 2 is chosen to be the most likely one as you computed in the previous question. Now what is the probability of every data point being chosen as the center for cluster 3? (The answer should contain 5 probabilities, each for every data point)

(0, 0): 0
(1, 2): 0.357
(2, 3): 0.571
(3, 1): 0.071
(4, 1): 0

4. Which data point is mostly liken chosen as the center for cluster 3?

(2, 3) is mostly likely chosen.

5. Assume the center for cluster 3 is also chosen to be the most likely one as you computed in the previous question. Now we finish the initialization for all 3 centers. List the data points that are classified into cluster 1, 2, 3 respectively.

cluster 1: (0, 0)
cluster 2: (1, 2), (2, 3)
cluster 3: (3, 1), (4, 1)

6. Based on the above clustering result, what's the new center for every cluster?

center for cluster 1: (0, 0)
center for cluster 2: (1.5, 2.5)
center for cluster 3: (3.5, 1)

7. According to the result of (ii) and (iv), explain how does k-means++ alleviate the local optimum issue due to initialization?

k-means++ tends to initialize new cluster centers with the data points that are far away from the existing centers, to make sure all of the initial cluster centers stay away from each other.

3. Consider a dataset with seven points $\{x_1, \ldots, x_7\}$. Given below are the distances between all pairs of points.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	5	3	1	6	2	3
x_2	5	0	4	6	1	7	8
x_3	3	4	0	4	3	5	6
x_4	1	6	4	0	7	1	2
x_5	6	1	3	7	0	8	9
x_6	2	7	5	1	8	0	1
x_7	3	8	6	2	9	1	0

Assume that $k = 2$, and the cluster centers are initialized to x_3 and x_6. Which of the following shows the two clusters formed at the end of the first iteration of k-means? Circle the correct option.

(a) $\{x_1, x_2, x_3, x_4\}$, $\{x_5, x_6, x_7\}$

(b) $\{x_2, x_3, x_5\}$, $\{x_1, x_4, x_6, x_7\}$

(c) $\{x_1, x_2, x_3, x_5\}$, $\{x_4, x_6, x_7\}$

(d) $\{x_2, x_3, x_4, x_7\}$, $\{x_1, x_5, x_6\}$

Solution: (b).

4 Naive Bayes

1. Consider the following data. It has 4 features $\mathbf{X} = (x_1, x_2, x_3, x_4)$ and 3 labels $y \in \{+1, 0, -1\}$. Assume that the probabilities $p(\mathbf{X}|y)$ and $p(y)$ are both Bernoulli distributions. Answer the questions that follow under the Naive Bayes assumption.

x_1	x_2	x_3	x_4	y
1	1	0	1	+1
0	1	1	0	+1
1	0	1	1	0
0	1	1	1	0
0	1	0	0	-1
1	0	0	1	-1
0	0	1	1	-1

1. Compute the Maximum Likelihood Estimates for $p(x_i = 1|y), \forall i \in \{1, 2, 3, 4\}$ and $\forall y \in \{+1, 0, -1\}$.

	$y = +1$	$y = 0$	$y = -1$
$x_1 = 1$			
$x_2 = 1$			
$x_3 = 1$			
$x_4 = 1$			

	$y = +1$	$y = 0$	$y = -1$
$x_1 = 1$	0.5	0.5	1/3
$x_2 = 1$	1	0.5	1/3
$x_3 = 1$	0.5	1	1/3
$x_4 = 1$	0.5	1	2/3

2. Compute the Maximum Likelihood Estimates for the prior probabilities $p(y = +1), p(y = 0), p(y = -1)$.

$p(y = +1) = \frac{2}{7}$, $p(y = 0) = \frac{2}{7}$ and $p(y = -1) = \frac{3}{7}$.

3. Use the values computed in the above two parts to classify the data point ($x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1$) as belonging to class $+1, 0$ or -1.

According to Naïve Bayes assumption, features are independent given y, thus we can write the conditional joint probability as

$$p(x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1) = p(x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1|y)p(y) \quad (1)$$
$$= p(y)\prod_{i=1}^{4} p(x_i = 1|y). \quad (2)$$

We calculate the probability given different values of y and pick the one with highest probability:

$$p(y = +1)\prod_{i=1}^{4} p(x_i = 1|y = +1) = \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{2}{7} = \frac{1}{28} \quad (3)$$
$$p(y = 0)\prod_{i=1}^{4} p(x_i = 1|y = 0) = \frac{1}{2} \cdot \frac{1}{2} \cdot 1 \cdot 1 \cdot \frac{2}{7} = \frac{1}{14} \quad (4)$$
$$p(y = -1)\prod_{i=1}^{4} p(x_i = 1|y = -1) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{3}{7} = \frac{2}{189} \quad (5)$$

Since $y = 0$ yields the largest value, we classify the data as $\hat{y} = 0$.

2. You are given a dataset of 10,000 students with their sex, height, and hair color. You are trying to build a machine learning classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

   - sex $\in$ {male,female}
   - height $\in$ [0,300] centimeters
   - hair $\in$ {brown, black, blond, red, green}
   - 3240 men in the data set
   - 6760 women in the data set

   Under only the assumptions necessary for Naïve Bayes (not the distributional assumptions you might naturally or intuitively make about the data set), answer True or False and provide a one sentence justification of your answer.

   1. **True or False:** Height is a continuous valued variable. Therefore, Naïve Bayes is not appropriate since it cannot handle continuous valued variables.

      False. Naïve Bayes can handle both continuous and discrete values as long as the appropriate distributions are used for conditional probabilities. For example, Gaussian for continuous and Bernoulli for discrete

   2. **True or False:** Since there aren't similar numbers of men and women in the data set, Naïve Bayes will have high test error.

      False. Since the data was randomly split, the same proportion of male and female will be in the training and testing sets. Thus this discrepancy will not affect testing error.

   3. **True or False:** $p(\texttt{height}|\texttt{sex}, \texttt{hair}) = p(\texttt{height}|\texttt{sex})$.

      True. This results from the conditional independence assumption required for Naïve Bayes.

4. **True or False:** $p(\texttt{height}, \texttt{hair}|\texttt{sex}) = p(\texttt{height}|\texttt{sex}) * p(\texttt{hair}|\texttt{sex})$.

True. This results from the conditional independence assumption required for Naïve Bayes.

5. Suppose you wish to learn $P(Y|X_1, X_2, X_3)$, where Y, X_1, X_2 and X_3 are all boolean-valued random variables. For the questions below, answer True or False and provide a one sentence justification for your answer.

   (a) **True or False:** In this case, a good choice for Naïve Bayes would be to implement a Gaussian Naïve Bayes classifier.

   Answer: False. Given the X_i are boolean, it is better to model $P(X_i|Y)$ with a Bernoulli rather than Gaussian distribution

   (b) **True or False:** To learn $P(Y|X_1, X_2, X_3)$ using Naïve Bayes, you must make conditional independence assumptions, including the assumption that Y is conditionally independent of X_1 given X_2.

   Answer: False. Naïve Bayes assume $(\forall i \neq j) X_i$ is conditionally independent of X_j given Y.

   (c) **True or False:** We can train Naïve Bayes using maximum likelihood estimates for each parameter, but not MAP estimates.

   False: MAP estimates are just MLE spiced up with priors on the parameters of $P(X_i|Y_j)$ (prior knowledge that we can inject into the model), so there's no reason we can't add it in.

6. Suppose we add a numeric, real-valued variable X_4 to our problem. Note we now have a mix of some discrete-valued X_i and one continuous X_i. Explain why we can no longer use Naïve Bayes, or if we can, how we would modify our original solution.

We can't use our original model because a Bernoulli distribution can't model the new data.

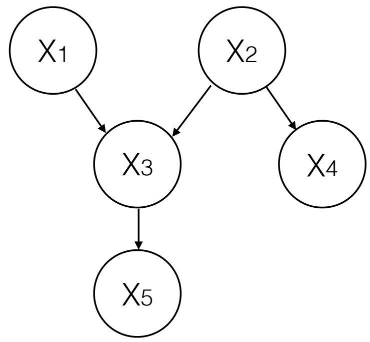
We must modify our solution so that a different Naïve Bayes model is trained on the continuous variables, using a different distribution than a Bernoulli one (i.e. a Gaussian), and then the result is a multiplication of the two.

Since our assumption is that each parameter is conditionally independent anyhow, we can multiply the results of the two models together safely.

5 Bayesian Networks

1. Consider the following Bayesian network.

   1. Determine whether the following conditional independencies are true.



(a) $X_1 \perp X_2 \mid X_3$?
Circle one: Yes No
No.

(b) $X_1 \perp X_4$?
Circle one: Yes No
Yes.

(c) $X_5 \perp X_2 \mid X_3$?
Circle one: Yes No
Yes.

2. Write out the joint probability in a form that utilizes as many independence/conditional independence assumptions contained in the graph as possible. Answer: $P(X_1, X_2, X_3, X_4, X_5) =$
$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_3)$

3. In a Bayesian network, if $X_1 \perp X_2$, then $X_1 \perp X_2|Y$ for every node Y in the graph.

   Circle one: True False

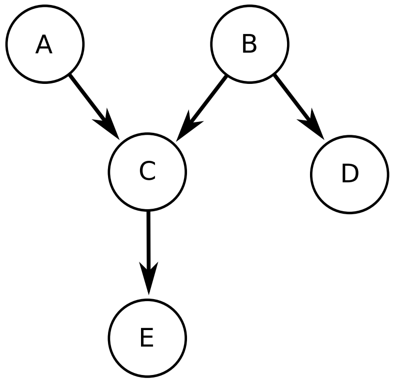
   False. Consider $X_1 \rightarrow Y \leftarrow X_2$.

4. In a Bayesian network, if $X_1 \perp X_2|Y$ for some node Y in the graph, it is always true that $X_1 \perp X_2$.

   Circle one: True False

   False. Consider $X_1 \leftarrow Y \rightarrow X_2$.

2. Consider the Bayesian network shown below for the following questions (a)-(f). Assume all variables are boolean-valued.



1. (Short answer) Write down the factorization of the joint probability $P(A, B, C, D, E)$ for the above graphical model, as a product of the five distributions associated with the five variables.

   $P(A, B, C, D, E) = P(A)P(B)P(C|A, B)P(D|B)P(E|C)$

2. **True or False**: Is C conditionally independent of D given B (i.e. is $(C \perp D)|B$)? True

3. **True or False**: Is A conditionally independent of D given C (i.e. is $(A \perp D)|C$)? False

4. **True or False**: Is A independent of B (i.e. is $A \perp B$)? True

5. Write an expression for $P(C = 1|A = 1, B = 0, D = 1, E = 0)$ in terms of the parameters of Conditional Probability Distributions associated with this graphical model.

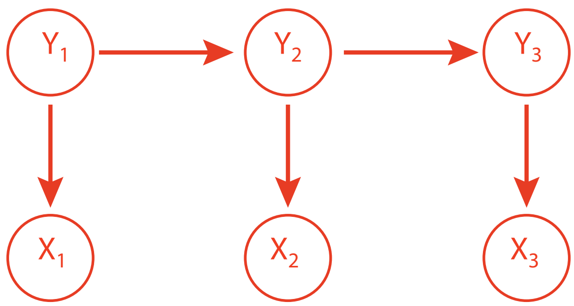
$$P(C = 1|A = 1, B = 0, D = 1, E = 0) = \frac{P(A = 1, B = 0, C = 1, D = 1, E = 0)}{\sum_{c=0}^{1} P(A = 1, B = 0, C = c, D = 1, E = 0)}$$

$$= \frac{P(A = 1)P(B = 0)P(C = 1|A = 1, B = 0)P(D = 1|B = 0)P(E = 0|C = 1)}{\sum_{c=0}^{1} P(A = 1)P(B = 0)P(C = c|A = 1, B = 0)P(D = 1|B = 0)P(E = 0|C = c)}$$

6 Hidden Markov Models

1. Recall that both the Hidden Markov Model (HMM) can be used to model sequential data with local dependence structures. In this question, let Y_t be the hidden state at time t, X_t be the observation at time t, $\mathbf{Y}$ be all the hidden states, and $\mathbf{X}$ be all the observations.

   1. Draw the HMM as a Bayesian network where the observation sequence has length 3 (i.e., $t = 1, 2, 3$), labelling nodes with Y_1, Y_2, Y_3 and X_1, X_2, X_3.



   2. Write out the factorized joint distribution of $P(\mathbf{X}, \mathbf{Y})$ using the independencies/-conditional independencies assumed by the HMM graph, using terms Y_1, Y_2, Y_3 and X_1, X_2, X_3.
      $P(\mathbf{X}, \mathbf{Y}) =$

      $P(\mathbf{X}, \mathbf{Y}) = P(Y_1)P(Y_2|Y_1)P(Y_3|Y_2)\prod_{t=1}^{3} P(X_t|Y_t)$

   3. True or False: In general, we should not include unobserved variables in a graphical model because we cannot learn anything useful about them without observations.
      True **False**

      False.

2. Consider an HMM with states $Y_t \in \{S_1, S_2, S_3\}$, observations $X_t \in \{A, B, C\}$ and parameters $\boldsymbol{\pi} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, transition matrix $\boldsymbol{B} = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$, and emission matrix $\boldsymbol{A} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$.

   1. What is $P(Y_5 = S_3)$?

$$\begin{aligned} & 1 - P(Y_5 = S_1) - P(Y_5 = S_2) \\ =& 1 - \frac{1}{16} - 4 \times \frac{1}{32} \\ =& \frac{13}{16} \end{aligned}$$

   2. What is $P(Y_5 = S_3 | X_{1:7} = AABCABC)$?

      0, since it is impossible for S_3 to output A.

3. Fill in the following table assuming the observation $AABCABC$. The α's are values obtained during the forward algorithm: $\alpha_t(i) = P(X_1, ..., X_t, Y_t = i)$.

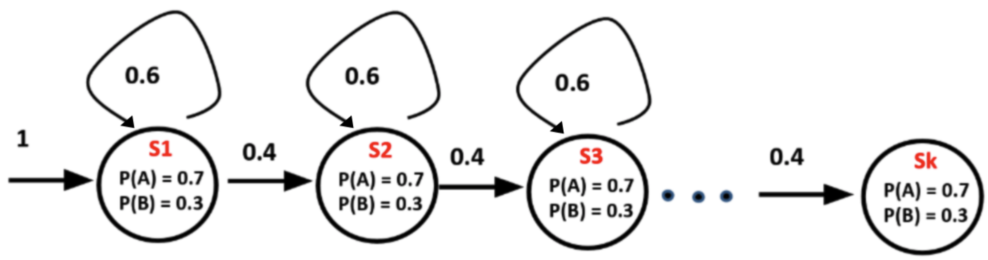
t	$\alpha_t(1)$	$\alpha_t(2)$	$\alpha_t(3)$
1			
2			
3			
4			
5			
6			
7			

t	$\alpha_t(1)$	$\alpha_t(2)$	$\alpha_t(3)$
1	$1/2$	0	0
2	$1/8$	$1/16$	0
3	$1/32$	0	$1/32$
4	0	$1/2^8$	$5/2^8$
5	0	$1/2^{10}$	0
6	0	0	$1/2^{12}$
7	0	0	$1/2^{13}$

4. Write down the sequence of $Y_{1:7}$ with the maximal posterior probability assuming the observation $AABCABC$. What is that posterior probability? $S_1 S_1 S_1 S_2 S_2 S_3 S_3$

posterior probability $= 1$

3. Consider the HMM in the figure below.



The HMM has k states $(s_1, ..., s_k)$. s_k is the terminal state. All states have the same emission probabilities (shown in the figure). The HMM always starts at s_1 as shown, and can move to either the next greater-number state or stay in the current state. Transition probabilities for all states except s_k are also the same as shown. More formally:

1. $P(Y_i = S_t \mid Y_{i-1} = S_{t-1}) = 0.4$
2. $P(Y_i = S_t \mid Y_{i-1} = S_t) = 0.6$
3. $P(Y_i = S_t \mid Y_{i-1} = S_j) = 0$ for all $j \in [k] \setminus \{t, t-1\}$

Once a run reaches s_k it outputs a symbol based on the s_k state emission probability and terminates.

1. Assume we observed the output AABAABBA from the HMM. Select all answers below that COULD be correct.

   ○ $k > 8$

   ○ $k < 8$

   ○ $k > 6$

   ○ $k < 6$

   ○ $k = 7$

   BCDE. It cannot be more that 8 since if it was we would have more than 8 values in the output.

2. Now assume that $k = 4$. Let $P('AABA')$ be the probability of observing AABA from a full run of the HMM. For the following equations, fill in the box with $>, <, =$ or ? (? implies it is impossible to tell).

   (a) $P('AAB')$ ☐ $P('BABA')$

   $<$, since we must have at least 4 outputs, $P('AAB') = 0$

   (b) $P('ABAB')$ ☐ $P('BABA')$

   $=$, since all states are the same, it does not matter where the Bs come from in terms of probability

   (c) $P('AAABA')$ ☐ $P('BBAB')$

   $>$, $P('BBAB') = 0.4^3 \times 0.3^4 \times 0.7$, and $P('AAABA')$ is a sum over 3 possibilities (we need to stay twice in one of the three states). So $P('AAABA') = 3 \times 0.4^3 \times 0.6 \times 0.7^4 \times 0.3$

7 Reinforcement Learning

7.1 Markov Decision Process

Environment Setup (may contain spoilers for Shrek 1)

Lord Farquaad is hoping to evict all fairytale creatures from his kingdom of Duloc, and has one final ogre to evict: Shrek. Unfortunately all his previous attempts to catch the crafty ogre have fallen short, and he turns to you, with your knowledge of Markov Decision Processes (MDP's) to help him catch Shrek once and for all.

Consider the following MDP environment where the agent is Lord Farquaad:

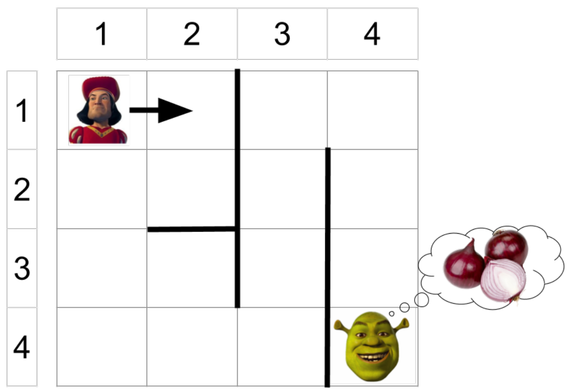


Figure 1: Kingdom of Duloc, circa 2001

Here's how we will define this MDP:

- S **(state space):** a set of states the agent can be in. In this case, the agent (Farquaad) can be in any location (row, col) and also in any orientation $\in \{N, E, S, W\}$. Therefore, state is represented by a three-tuple (row, col, dir), and $S =$ all possible of such tuples. Farquaad's start state is $(1, 1, E)$.
- A **(action space):** a set of actions that the agent can take. Here, we will have just three actions: turn right, turn left, and move forward (turning does not change row or col, just dir). So our action space is $\{R, L, M\}$. Note that Farquaad is debilitatingly short, so he cannot travel through (or over) the walls. Moving forward when facing a wall results in no change in state (but counts as an action).
- $R(s, a)$ **(reward function):** In this scenario, Farquaad gets a reward of 5 by moving into the swamp (the cell containing Shrek), and a reward of 0 otherwise.
- $p(s'|s, a)$ **(transition probabilities):** We'll use a deterministic environment, so this will bee 1 if s' is reachable from s and by taking a, and 0 if not.

1. What are $|S|$ and $|A|$ (size of state space and size of action space)?

   $|S| = 4$ rows $\times$ 4 columns $\times$ 4 orientations $= 64$
   $|A| = |\{R, L, M\}| = 3$

2. Why is it called a "Markov" decision process? (Hint: what is the assumption made with p?)

   $p(s'|s, a)$ assumes that s' is determined only by s and a (and not any other previous states or actions).

3. What are the following transition probabilities?

$$p((1,1,N)|(1,1,N), M) =$$
$$p((1,1,N)|(1,1,E), L) =$$
$$p((2,1,S)|(1,1,S), M) =$$
$$p((2,1,E)|(1,1,S), M) =$$

$$p((1,1,N)|(1,1,N), M) = 1$$
$$p((1,1,N)|(1,1,E), L) = 1$$
$$p((2,1,S)|(1,1,S), M) = 1$$
$$p((2,1,E)|(1,1,S), M) = 0$$

4. Given a start position of $(1, 1, E)$ and a discount factor of $\gamma = 0.5$, what is the expected discounted future reward from $a = R$? For $a = L$? (Fix $\gamma = 0.5$ for following problems).

   For $a = R$ we get $R_R = 5 * (\frac{1}{2})^{16}$ (it takes 17 moves for Farquaad to get to Shrek, starting with R, M, M, M, L...)

   For $a = L$, this is a bad move, and we need another move to get back to our original orientation, from which we can go with our optimal policy. So the reward here is:

   $R_L = (\frac{1}{2})^2 * R_R = 5 * (\frac{1}{2})^{18}$

5. What is the optimal action from each state, given that orientation is fixed at E? (if there are multiple options, choose any)

R	R	M	R
R	R	L	R
M	R	L	R
M	M	L	-

(some have multiple options, I just chose one of the possible ones)

6. Farquaad's chief strategist (Vector from Despicable Me) suggests that having $\gamma = 0.9$ will result in a different set of optimal policies. Is he right? Why or why not?

Vector is wrong. While the reward quantity will be different, the set of optimal policies does not change. (it is now $5 * (\frac{9}{10})^{16}$) (one can only assume that Lord Farquaad and Vector would be in kahoots: both are extremely nefarious!)

7. Vector then suggests the following setup: $R(s, a) = 0$ when moving into the swamp, and $R(s, a) = -1$ otherwise. Will this result in a different set of optimal policies? Why or why not?

It will not. While the reward quantity will be different, the set of optimal policies does not change. (Farquaad will still try to minimize the number of steps he takes in order to reach Shrek)

8. Vector now suggests the following setup: $R(s, a) = 5$ when moving into the swamp, and $R(s, a) = 0$ otherwise, but with $\gamma = 1$. Could this result in a different optimal policy? Why or why not?

This will change the policy, but not in Lord Farquaad's favor. He will no longer be incentivized to reach Shrek quickly (since $\gamma = 1$). The optimal reward from each state is the same (5) and therefore each action from each state is also optimal. Vector really should have taken 10-301/601...

9. Surprise! Elsa from Frozen suddenly shows up. Vector hypnotizes her and forces her to use her powers to turn the ground into ice. The environment is now stochastic: since the ground is now slippery, when choosing the action M, with a 0.2 chance, Farquaad will slip and move two squares instead of one. What is the expected future-discounted rewards from $s = (2, 4, S)$?

   Recall that $R_{exp} = max_a E[R(s, a) + \gamma R_{s'}]$

   (notation might be different than in the notes, but conceptually, our reward is the best expected reward we can get from taking any action a from our current state s.)

   In this case, our best action is obviously to move forward. So we get

   R_{exp} = (expected value of going two steps) + (expected value of going one step)

   $E[2_{steps}] = p((4, 4, S)|(2, 4, S), M) \times R((4, 4, S), (2, 4, S), M) = 0.2 \times 5 = 1$

   $E[1_{step}] = p((4, 3, S)|(2, 4, S), M) \times (R((4, 3, S), (2, 4, S), M) + \gamma R_{(4,3,S)})$

   where $R_{(4,3,S)}$ is the expected reward from $(4, 3, S)$. Since the best reward from here is obtained by choosing $a = M$, and we always end up at Shrek, we get

   $E[1_{step}] = 0.8 \times (0 + \gamma \times 5) = 0.8 \times 0.5 \times 5 = 2$

   giving us a total expected reward of $R_{exp} = 1 + 2 = 3$

   (I will be very disappointed if this is not the plot of Shrek 5)

7.2 Value and Policy Iteration

1. **Select all that apply:** Which of the following environment characteristics would increase the computational complexity per iteration for a value iteration algorithm? Choose all that apply:

   ☐ Large Action Space

   ☐ A Stochastic Transition Function

   ☐ Large State Space

   ☐ Unknown Reward Function

   ☐ None of the Above

   A and C (state space and action space). The computational complexity for value iteration per iteration is $O(|A||S|^2)$

   B is NOT correct. The time complexity is $O(|A||S|^2)$ for both stochastic and deterministic transition (review the lecture slides).

2. **Select all that apply:** Which of the following environment characteristics would increase the computational complexity per iteration for a policy iteration algorithm? Choose all that apply:

   ☐ Large Action Space

   ☐ A Stochastic Transition Function

   ☐ Large State Space

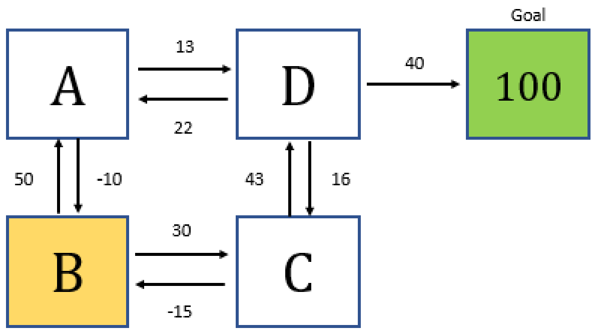
   ☐ Unknown Reward Function

   ☐ None of the Above

   A and C again. The computational complexity for policy iteration per iteration is $O(|A||S|^2 + |S|^3)$

   Again, B is NOT correct.

3. In the image below is a representation of the game that you are about to play. There are 5 states: A, B, C, D, and the goal state. The goal state, when reached, gives 100 points as reward (that is, you can assume $R(\texttt{D}, \texttt{right}) = 140$). In addition to the goal's points, you also get points by moving to different states. The amount of points you get are shown next to the arrows. You start at state B. To figure out the best policy, you use asynchronous value iteration with a decay (γ) of 0.9. You should initialize the value of each state to 0.



(a) When you first start playing the game, what action would you take (up, down, left, right) at state B?

Up

(b) What is the total reward at state B at this time?

50 (immediate reward of 50, and future reward (value at state A) starts at 0)

(c) Let's say you keep playing until your total values for each state has converged. What action would you take at state B?

C

(d) What is the total reward at state B at this time?

182.1 (30 from the immediate action, and $43 * 0.9 + (100 + 40) * 0.9^2 = 152.1$ from the future reward (value at state C))

4. **Select one:** Let $V_k(s)$ indicate the value of state s at iteration k in (synchronous) value iteration. What is the relationship between $V_{k+1}(s)$ and $\sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')]$, for any $a \in A$? Indicate the most restrictive relationship that applies. For example, if $x < y$ always holds, use $<$ instead of $\leq$. Selecting ? means it's not possible to assign any true relationship. Assume $R(s,a,s') \geq 0 \ \forall s, s' \in S,\ a \in A$.

   $V_{k+1}(s) \ \Box \ \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')]$

   ◯ $=$

   ◯ $<$

   ◯ $>$

   ◯ $\leq$

   ◯ $\geq$

   ◯ ?

   E

7.3 Q-Learning

1. For the following true/false, circle one answer and provide a one-sentence explanation:

   (a) One advantage that Q-learning has over Value and Policy iteration is that it can account for non-deterministic policies.

   Circle one: True False

   False. All three methods can account for non-deterministic policies

   (b) You can apply Value or Policy iteration to any problem that Q-learning can be applied to.

   Circle one: True False

   False. Unlike the others, Q-learning doesn't need to know the transition probabilities (p(s' | s, a)), or the reward function (r(s,a)) to train. This is its biggest advantage.

   (c) Q-learning is guaranteed to converge to the true value Q* for a greedy policy.

   Circle one: True False

   False. Q-learning converges only if every state will be explored infinitely. Thus, purely exploiting policies (e.g. greedy policies) will not necessarily converge to Q*, but rather to a local optimum.

2. For the following parts of this problem, recall that the update rule for Q-learning is:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \left(q(\mathbf{s}, a; \mathbf{w}) - (r + \gamma \max_{a'} q(\mathbf{s}', a'; \mathbf{w}) \right) \nabla_{\mathbf{w}} q(\mathbf{s}, a; \mathbf{w})$$

   (a) From the update rule, let's look at the specific term $X = (r + \gamma \max_{a'} q(\mathbf{s}', a'; \mathbf{w}))$ Describe in English what is the role of X in the weight update.

   Estimate of true total return (Q*(s,a)). This may get multiple answers, so grade accordingly

   (b) Is this update rule synchronous or asynchronous?

   Asynchronous

   (c) A common adaptation to Q-learning is to incorporate rewards from more time steps into the term X. Thus, our normal term $r_t + \gamma * max_{a_{t+1}} q(s_{t+1}, a_{t+1}; w)$ would become $r_t + \gamma * r_{t+1} + \gamma^2 \max_{a_{t+2}} q(\mathbf{s}_{t+2}, a_{t+2} : \mathbf{w})$ What are the advantages of using more rewards in this estimation?

   Incorporating rewards from multiple time steps allows for a more "realistic" estimate of the true total reward, since a larger percentage of it is from real experience. It can help with stabilizing the training procedure, while still allowing training at each time step (bootstrapping). This type of method is called N-Step Temporal Difference Learning.

3. **Select one:** Let $Q(s, a)$ indicate the estimated Q-value of state-action pair $(s, a) \in |S| \times |A|$ at some point during Q-learning. Suppose you receive reward r after taking action a at state s and arrive at state s'. Before updating the Q values based on this experience, what is the relationship between $Q(s, a)$ and $r + \gamma \max_{a' \in A} Q(s', a')$? Indicate the most restrictive relationship that applies. For example, if $x < y$ always holds, use $<$ instead of $\leq$. Selecting ? means it's not possible to assign any true relationship.

   $Q(s, a) \square r + \gamma \max_{a'} Q(s', a')$

   ○ $=$

   ○ $<$

   ○ $>$

   ○ $\leq$

   ○ $\geq$

   ○ ?

   F

4. During standard (not deep) Q-learning, you get reward r after taking action $North$ from state A and arriving at state B. You compute the sample $r + \gamma Q(B, South)$, where $South = \arg\max_a Q(B, a)$.

   Which of the following Q-values are updated during this step? (Select all that apply)

   ◯ Q(A, North)

   ◯ Q(A, South)

   ◯ Q(B, North)

   ◯ Q(B, South)

   ◯ None of the above

   A

5. In general, for Q-Learning (standard/tabular Q-learning, not approximate Q-learning) to converge to the optimal Q-values, which of the following are true?

   (a) **True or False:** It is necessary that every state-action pair is visited infinitely often.

   ◯ True

   ◯ False

   (b) **True or False:** It is necessary that the discount γ is less than 0.5.

   ◯ True

   ◯ False

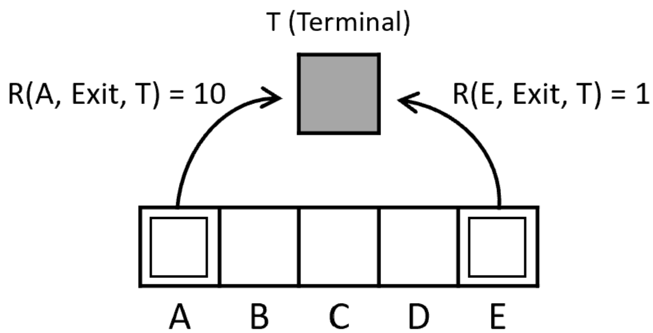
   (c) **True or False:** It is necessary that actions get chosen according to $\arg\max_a Q(s, a)$.

   ◯ True

   ◯ False

   a) **True**: In order to ensure convergence in general for Q learning, this has to be true. In practice, we generally care about the policy, which converges well before the values do, so it is not necessary to run it infinitely often. b) **False**: The discount factor must be greater than 0 and less than 1, not 0.5. c) **False**: This would actually do rather poorly, because it is purely exploiting based on the Q-values learned thus far, and not exploring other states to try and find a better policy.

6. Consider training a robot to navigate the following grid-based MDP environment.



- There are six states, A, B, C, D, E, and a terminal state T.
- Actions from states B, C, and D are Left and Right.
- The only action from states A and E is Exit, which leads deterministically to the terminal state

The reward function is as follows:

- $R(A, Exit, T) = 10$
- $R(E, Exit, T) = 1$
- The reward for any other tuple (s, a, s') equals -1

Assume the discount factor is 1. When taking action Left, with probability 0.8, the robot will successfully move one space to the left, and with probability 0.2, the robot will move one space in the opposite direction. When taking action Right, with probability 0.8, the robot will successfully move one space to the right, and with probability 0.2, the robot will move one space in the opposite direction. Run synchronous value iteration on this environment for two iterations. Begin by initializing the value of all states to zero.

Write the value of each state after the first ($k = 1$) and the second ($k = 2$) iterations. Write your values as a comma-separated list of 6 numerical expressions in the alphabetical order of the states, specifically $V(A), V(B), V(C), V(D), V(E), V(T)$. Each of the six entries may be a number or an expression that evaluates to a number. Do not include any max operations in your response.

(a) $V_1(A), V_1(B), V_1(C), V_1(D), V_1(E), V_1(T)$ (Values for 6 states):

$10, -1, -1, -1, 1, 0$

(b) $V_2(A), V_2(B), V_2(C), V_2(D), V_2(E), V_2(T)$ (values for 6 states):

$10, 6.8, -2, -0.4, 1, 0$

(c) What is the resulting policy after this second iteration? Write your answer as a comma-separated list of three actions representing the policy for states, B, C, and D, in that order. Actions may be Left or Right.

$\pi(B), \pi(C), \pi(D)$ based on V_2 :

Left, Left, Right

8 Ensemble Methods

1. Consider a random forest ensemble consisting of 5 decision trees DT1, DT2 ... DT5 that has been trained on a dataset consisting of 7 samples. Each tree has been trained on a random subset of the dataset. The following table represents the predictions of each tree on its out-of-bag samples.

Tree	Sample Number	Prediction	Actual
DT1	6	No	Yes
DT1	7	No	Yes
DT2	2	No	No
DT3	1	No	No
DT3	2	Yes	No
DT3	4	Yes	Yes
DT4	2	Yes	No
DT4	7	No	Yes
DT5	3	Yes	Yes
DT5	5	No	No

1. What is the OOB error of the above random forest classifier?

   OOB is the average error of the table which is 0.5.

2. In the above random forest classifier, which Decision tree(s) will be given the highest weight in inference? If there are multiple trees, mention them all

   DT1, DT2, DT3, DT4, DT5. Random Forests do unweighted sums of the individual tree predictions

3. To reduce the error of each individual decision tree, Neural uses all the features to train each tree. How would this impact the generalisation error of the random forest?

   ◯ The generalisation error would decrease as each tree has lower generalisation error

   ◯ The generalisation error would increase as each tree has insufficient training data

   ◯ The generalisation error would increase as the trees are highly correlated

   The generalisation error would increase as the trees are highly correlated

2. In the AdaBoost algorithm, if the final hypothesis makes no mistakes on the training data, which of the following is correct?

   Select all that apply:

   ☐ Additional rounds of training can help reduce the errors made on unseen data.

   ☐ Additional rounds of training have no impact on unseen data.

   ☐ The individual weak learners also make zero error on the training data.

   ☐ Additional rounds of training always leads to worse performance on unseen data.

   A. AdaBoost is empirically robust to overfitting and the testing error usually continues to reduce with more rounds of training.

3. **True or False:** In AdaBoost weights of the misclassified examples go up by the same multiplicative factor.

   ◯ True

   ◯ False

   True, follows from the update equation.

Round	$D_t(A)$	$D_t(B)$	$D_t(C)$	$D_t(D)$	$D_t(E)$	$D_t(F)$
1	?	?	$\frac{1}{6}$	?	?	?
2	?	?	?	?	?	?
...						
219	?	?	?	?	?	?
220	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{7}{14}$	$\frac{1}{14}$	$\frac{2}{14}$	$\frac{2}{14}$
221	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{7}{20}$	$\frac{1}{20}$	$\frac{1}{4}$	$\frac{1}{10}$
...						
3017	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	0
...						
8888	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

4. In the last semester, someone used AdaBoost to train some data and recorded all the weights throughout iterations but some entries in the table are not recognizable. Clever as you are, you decide to employ your knowledge of Adaboost to determine some of the missing information.

   Below, you can see part of table that was used in the problem set. There are columns for the Round # and for the weights of the six training points (A, B, C, D, E, and F) at the start of each round. Some of the entries, marked with "?", are impossible for you to read.

   In the following problems, you may assume that non-consecutive rows are independent of each other, and that a classifier with error less than $\frac{1}{2}$ was chosen at each step.

   1. The weak classifier chosen in Round 1 correctly classified training points A, B, C, and E but misclassified training points D and F. What should the updated weights have been in the following round, Round 2? Please complete the form below.

Round	$D_2(A)$	$D_2(B)$	$D_2(C)$	$D_2(D)$	$D_2(E)$	$D_2(F)$
2						

$\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{8}, \frac{1}{4}$

   2. During Round 219, which of the training points (A, B, C, D, E, F) must have been misclassified, in order to produce the updated weights shown at the start of Round 220? List all the points that were misclassified. If none were misclassified, write 'None'. If it can't be decided, write 'Not Sure' instead.

Not sure

3. You observes that the weights in round 3017 or 8888 (or both) cannot possibly be right. Which one is incorrect? Why? Please explain in one or two short sentences.

   ◯ Round 3017 is incorrect.

   ◯ Round 8888 is incorrect.

   ◯ Both rounds 3017 and 8888 are incorrect.

   C. 3017: weight cannot be 0; 8888: sum of weights should be 1.

5. What condition must a weak learner satisfy in order for boosting to work?
   Short answer:

   The weak learner must classify above chance performance.

6. After an iteration of training, AdaBoost more heavily weights which data points to train the next weak learner? (Provide an intuitive answer with no math symbols.)
   Short answer:

   The data points that are incorrectly classified by weak learners trained in previous iterations are more heavily weighted.

7. **Extra credit** Do you think that a deep neural network is nothing but a case of boosting? Why or why not? Impress us.
   Answer:

   Both viewpoints can be argued. One may view passing a linear combination through a nonlinear function as a weak learner (e.g., logistic regression), and that the deep neural network corrects for errors made by these weak learners in deeper layers. Then again, every layer of the deep neural network is optimized in a global fashion (i.e., all weights are updated simultaneously) to improve performance, which could possibly capture dependencies which boosting could not.