# Solutions

| | |
|---|---|
| **10-601 Machine Learning** | **Name:** |
| **Summer 2025** | **Andrew ID:** |
| **Practice Problems** | **Room:** |
| **Updated: June 16, 2025** | **Seat:** |
| **Time Limit: N/A** | **Exam Number:** |

**Instructions:**

- Verify your name and Andrew ID above.
- This exam contains 42 pages (including this cover page).
  The total number of points is 0.
- Clearly mark your answers in the allocated space. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- Look over the exam first to make sure that none of the 42 pages are missing.
- No electronic devices may be used during the exam.
- Please write all answers in pen or *darkly* in pencil.
- You have N/A to complete the exam. Good luck!

| Question | Points |
|---|---|
| 1. Learning Theory | 0 |
| 2. CNNs and RNNs | 0 |
| 3. Language Modeling, Attention & Transformers | 0 |
| 4. Pre-training, Fine-tuning & In-context Learning | 0 |
| 5. *k*-means | 0 |
| 6. Principal Component Analysis | 0 |
| 7. Reinforcement Learning | 0 |
| 8. Ensemble Methods | 0 |
| Total: | 0 |

# Instructions for Specific Problem Types

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

**Select One:** Who taught this course?

● Matt Gormley

○ 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 instructors for this course?

■ Matt Gormley

■ Henry Chai

□ 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 the instructors for this course?

■ Matt Gormley

■ Henry Chai

☒ 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-~~6~~301 |

# 1 Learning Theory (0 points)

1.1. Consider a binary classification problem with an unknown distribution over data points $D$ and an unknown target function $c^* : \mathcal{R}^d \mapsto \pm 1$. For any sample of points $S$ drawn from $D$, answer whether the following statements are true or false, and provide a brief justification.

(a) **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 $\epsilon$ 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.

(b) **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.

1.2. **Short answer:** Briefly describe the difference between the realizable case and agnostic case of PAC learning.

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.

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.

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

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

________________________________________

________________________________________

________________________________________

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

(b) **Select one:** What is the VC dimension of $\mathcal{H}$?

○ 0

○ 1

○ 2

C

(c) **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).

□

$2k$

1.5. 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 for her given model choice, $\mathcal{H}$, she is in the finite, agnostic case.

Now she wants to know how changing certain values will change the sample complexity i.e., the number of labeled training data points required to satisfy the PAC criterion:

$$P\left(|R(h) - \hat{R}(h)| \leq \epsilon\right) \geq 1 - \delta \; \forall \; h \in \mathcal{H}$$

where $R(h)$ and $\hat{R}(h)$ are the expected and empirical risks respectively.

For each of the following changes, determine whether the sample complexity will increase, decrease, or stay the same.

(a) **Select one:** Using a simpler model (decreasing $|\mathcal{H}|$)

○ Sample complexity will increase

○ Sample complexity will decrease

○ Sample complexity will stay the same

B

(b) **Select one:** 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

(c) **Select one:** Decreasing $\delta$

○ Sample complexity will increase

○ Sample complexity will decrease

○ Sample complexity will stay the same

A

(d) **Select one:** Decreasing $\epsilon$

○ Sample complexity will increase

○ Sample complexity will decrease

○ Sample complexity will stay the same

A

# 2 CNNs and RNNs (0 points)

2.1. Let's begin by considering some of the high-level components of a convolutional filter along with the basic motivation.

(a) **Short answer:** What is a convolutional filter?

A matrix or tensor of weights that is slid over an image tensor. At each position that is dictacted by hyperparameters such as padding/stride/kernel-size, we perform elementwise multiplication of the kernel with the underlying part of the image tensor and sum the resulting entries to obtain an output value.

(b) **Short answer:** Why do we need stride, and what benefits/tradeoffs might different values of stride have on the output?

The stride defines how many pixels the kernel moves with each step as it passes over the rows/columns of the image tensor. Larger values of stride can allow you to reduce the output dimensionality which could combat overfitting along with reducing computational power. The downside however is that you lose more and more information with larger values of stride, which could limit the upside of your model's accuracy. Very small stride (i.e. stride = 1) preserves the size of the input image and can identify more fine-grained features, but comes with greater computational cost.

(c) **Short answer:** What functionality does padding add to the convolutional layer and why might we want to use it?

Padding surrounds the image tensor with rows/columns of zeros. By adding an appropriate amount, we can ensure that the output shape is the same as the input shape, and allow every pixel to be included in the convolution. Furthermore, padding helps filters focus on the corner pixels just as much as middle pixels by making the filter pass over the corner pixels multiple times as opposed to just once.

2.2. Consider the following image, filter, and output shape.

$$X = \begin{bmatrix} 1 & 0 & -2 & 3 & 4 & 1 \\ 2 & 9 & 5 & 6 & 0 & -1 \\ 0 & -3 & 1 & 3 & 4 & 4 \\ 6 & 5 & 2 & 0 & 6 & 8 \\ -5 & 4 & -3 & 1 & 3 & -2 \\ 4 & 1 & 2 & 8 & 9 & 7 \end{bmatrix} \qquad F = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}$$

The shape of this particular $Y$ is the result of using no padding and a stride of 1.

(a) **Numerical answer:** Suppose we decide that, instead of having our output shape be $(4, 4)$, we want a slightly smaller, $(3, 3)$ image as output. In order for this to happen, what is the smallest combination of stride and padding that would work?

s=2, p=1. Notice that in this setting the kernel is never actually overlaid over the rightmost column or bottom most row of padding, since doing so would cause the kernel to go out of bounds.

(b) **Math:** Let's make this a bit more general. Suppose our original image has shape $(a, a)$, and we want the shape of our final image to be $(b, b)$, where $b \leq a$. Furthermore the shape of the filter is $(k, k)$, the stride length is $s$, and the padding is $p$. Express $b$ in terms of all defined variables.

$b = \lfloor ((a + 2p - k)/s) + 1 \rfloor$

2.3. Consider parameter sharing in a CNN.

(a) **Math:** Consider the setup from 2.2.(b), with the same variables representing the same quantities. How many parameters do we learn *with* parameter sharing?

□

$k^2$

(b) **Math:** Consider the setup from 2.2.(b), with the same variables representing the same quantities. How many parameters do we learn *without* parameter sharing?

□

$k^2b^2$

(c) **Short answer:** Suppose that this CNN is being used for a prediction task in a subject you have prior knowledge in - for instance, suppose you are being asked to classify images of cars, and you know that each image contains a side view of the car with the front of the car facing right and each car is roughly the same size. In this scenario, would parameter sharing be appropriate, disregarding computational constraints? Why or why not?

________________________________________

________________________________________

________________________________________

Here is it more appropriate not to share parameters. Because we know that all filters placed on the same area within a given image will be looking for the same thing (e.g. bumpers, rear windows, front wheels, etc.), having one set of parameters try to identify each of these features is not necessary when we can have a set of filters, each responsible identifying a different segment/part of the car.

2.4. **Short answer:** What is the difference between upsampling and downsampling in

CNNs? What are the appropriate scenarios to use them?

Downsampling: the goal is to *reduce* the output dimensionality. Some common methods include pooling functions (e.g. maxpooling, mean pooling). These are appropriate for larger images where you want to reduce the compute time/mitigate overfitting by making the input to future convolutions less complex.

Upsampling: the goal is to *increase* the output dimensionality. Upsampling is generally used when you want the output of a convolution to be larger e.g. match the input image in dimensionality. For instance you might be assigning a label to each pixel of the output and matching it with the input.

2.5. **Select all that apply**: In which of the following settings is it more appropriate to use an RNN over a CNN?

- □ Speech recognition
- □ Facial recognition
- □ Music composition
- □ Autocorrect system
- □ None of the above

A, C, D

2.6. **True or False**: RNN's are helpful in analyzing time series data. Briefly justify your answer in 1-2 sentences.

- ○ True
- ○ False

True. RNNs can handle sequential information, and are also able to incorporate information from previous time steps into the current time step. In addition, RNNs can process inputs of variable lengths, which is helpful for time series data.

# 3 Language Modeling, Attention & Transformers (0 points)

3.1. **Short answer:** Your friend is painfully running a Transformer model on their a laptop, and their computer seems to be struggling with the large vocabulary size. To solve the problem, they try tokenizing letters instead of words: this way, the vocabulary only contains 26 tokens. Explain why this is not a good idea, and suggest an alternative method that would reduce vocabulary size.

Language Models work because specific tokens, and their associated embeddings, represent units of meaning that can interact with each other. A letter alone does not contain any meaning, so a Transformer architecture that tokenizes letters will not be able to extract meaning out of a sentence. At most, it will be able to identify and reproduce common phonetic patterns or small words.

Possible alternatives to reduce vocabulary size include (but are not restricted to):

- Cutting the tail of the word distribution (i.e. simply removing a lot of low-frequency words from the vocabulary)
- Sub-word tokenization

3.2. **Select one:** Suppose a multi-head self-attention mechanism with 4 heads is applied to 20-token-long sequences where each token has an embedding dimensionality of 5. If we use scaled dot product attention with 10-dimensional keys and queries (so $d_k = d_q = 10$) and 3-dimensional values (so $d_v = 3$), what is the total dimensionality of the output for one sequence after concatenation?

○ $4 * 20 * 3 = 240$

○ $4 * 20 * 10 = 800$

○ $4 * 20 * 5 = 400$

○ $4 * 5 * 3 = 60$

○ $4 * 5 * 10 = 200$

A

3.3. **Short answer:** In 1-2 concise sentences, briefly describe the importance of multi-head attention over single head attention.

With multiple attention heads, the model can focus on different aspects of the input, enhancing its ability to understand complex dependencies in tasks like natural language processing. Each head will pay attention to a distinct input element individually, the model does a better job of capturing positional detail. Since it can also be parallelized it also boosts computational efficiency which makes transformers more versatile.

3.4. **Ordering:** In lecture, we saw that a transformer consists of many components/steps. Order the following items based on the order in which they are applied to an input to a transformer, starting from raw text. Specify your answer by writing a number next to each item, beginning with 1; if an item is not a part of the Transformer architecture, write ∅ in the associated space.

- ____ Embedding
- ____ Pooling
- ____ Feed-forward Neural Network
- ____ Tokenization
- ____ Positional Encoding
- ____ Layer Norm
- ____ Multi-Head Attention

2 Embedding
∅ Pooling
6 Feed-forward Neural Network
1 Tokenization
3 Positional Encoding
5 Layer Norm
4 Multi-Head Attention

# 4 Pre-training, Fine-tuning & In-context Learning (0 points)

4.1. **Short answer:** Why is unsupervised pre-training considered a plausible approach to improving the training of deep neural networks? Briefly explain its underlying intuition.

Unsupervised pre-training trains each layer of the network iteratively using the training dataset by minimizing the reconstruction error. This allows each layer progressively learns meaningful representations, capturing increasingly abstract features as data moves through the network. This ensures that each layer starts with a well-initialized foundation, reducing the risk of vanishing gradients and guiding the network closer to an optimal solution.

4.2. **True or False:** In-context learning selects a subset of the training data points to make a prediction on some test data point and adjusts the LLM's parameters to maximize the likelihood of the selected subset.

○ True

○ False

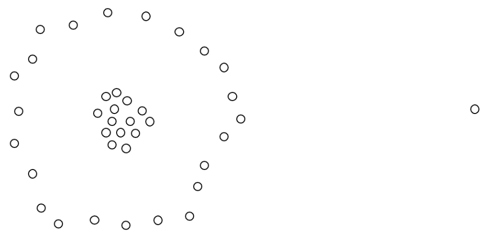
False, ICL does not adjust any parameters.

4.3. **Short answer:** A pre-trained language model is to be deployed for two tasks: medical report analysis and sentiment classification of a product review. For medical report analysis, the user will ask the model follow-up questions about a report it is prompted with. For sentiment classification, the model is prompted with a review for a product and must classify the review as either {positive, negative or neutral}. For each task, decide whether fine-tuning or in-context learning would be more appropriate and justify your choice with specific reasons related to the nature of each task.

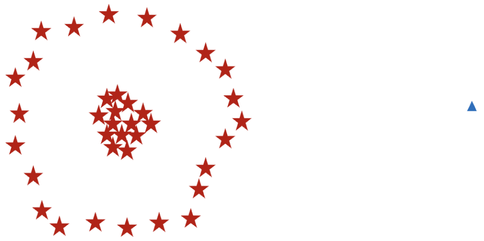
For medical report analysis, fine-tuning would be more appropriate due to the specialized and technical nature of medical reports, requiring the model to adapt to specific medical terminologies and concepts. In contrast, for sentiment classification, in-context learning might be better suited as it allows the model to leverage its pre-trained knowledge of language and style. The model doesn't need extensive specialized knowledge to know if a review is positive or negative, it should already know what language is associated with these concepts.

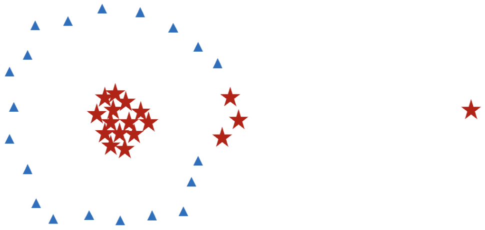
# 5 *k*-means (0 points)

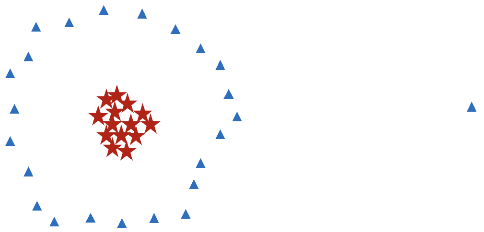
5.1. **Select all that apply:** Consider performing $k$-means clustering using Lloyd's method on the following dataset with $k = 2$ (note the outlier on the far right):

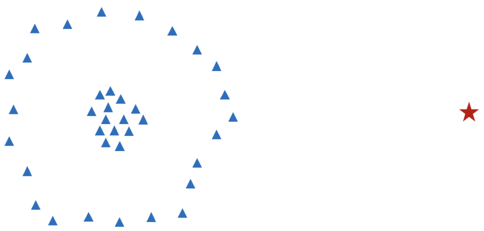


Which of the following could be the final clustering at convergence? In the options below, cluster membership is denoted by color and shape: red stars or blue triangles.

□ 

□ 

□ 

□ 

□ None of the above

A and D

5.2. Answer whether the following statements are true or false and if space is provided, provide a brief justification.

(a) **True or False:** k-means can always converge to the global optimum.

○ True

○ False

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

(b) **True or False:** k-means is sensitive to outliers.

○ True

○ False

True. 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.

(c) Consider the following generalized version of the $k$-means++ algorithm:

- Choose $\mathbf{c}_1$ at random.
- For $j = 2, \cdots, K$
  - Pick $\mathbf{c}_j$ among $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}$ according to the distribution

$$P(\mathbf{c}_j = \mathbf{x}^{(i)}) \propto \min_{j'<j} \|\mathbf{x}^{(i)} - \mathbf{c}_{j'}\|^{\alpha}$$

**True or False:** The version of $k$-means++ presented in lecture is equivalent to this version with $\alpha = 2$.

○ True

○ False

True.

(d) **True or False:** When $\alpha = 0$, this algorithm is equivalent to farthest point initialization.

○ True

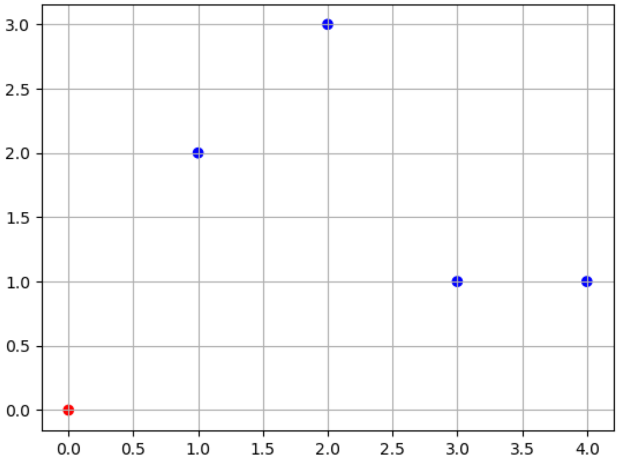
○ False

False: when $\alpha = 0$, this algorithm is equivalent to Lloyd's method where cluster centers are chosen randomly.

5.3. In $k$-means, random initialization of the cluster centers tends to lead to a local optimum with poor performance. To alleviate this issue, instead of initializing all of the centers completely randomly, we decide to use $k$-means++:

```
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
            $\mu^{(j)} \sim \text{Categorical}(\{x^{(i)}\}_{i=1}^N, \mathbf{p}_{1:N})$
```

Suppose we have 5 data points: (0, 0), (1, 2), (2, 3), (3, 1), (4, 1) and (0, 0) is randomly chosen to be the center of cluster 1 (shown in red in the figure below):



(a) **Numerical answer:** What is the probability of every data point being chosen as the center for cluster 2? Your answer should contain 5 probabilities, one for each data point.

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

(b) **Select one:** Which data point is most likely to be chosen as the center for cluster 2?

○ (0, 0)

○ (1, 2)

○ (2, 3)

○ (3, 1)

○ (4, 1)

E. (4, 1)

(c) **Numerical answer:** Assume the center for cluster 2 is chosen to be the most likely one. What is the probability of every data point being chosen as the center for cluster 3? Your answer should again contain 5 probabilities, one for each data point.

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

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

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?

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |
|---|---|---|---|---|---|---|---|
| $x_1$ | 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 |

○ $\{x_1, x_2, x_3, x_4\}$, $\{x_5, x_6, x_7\}$

○ $\{x_2, x_3, x_5\}$, $\{x_1, x_4, x_6, x_7\}$

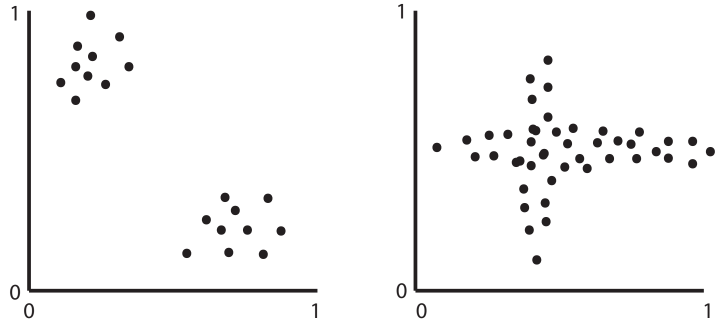
○ $\{x_1, x_2, x_3, x_5\}$, $\{x_4, x_6, x_7\}$

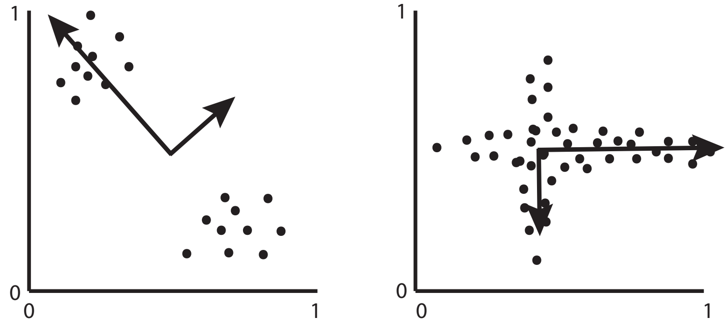
○ $\{x_2, x_3, x_4, x_7\}$, $\{x_1, x_5, x_6\}$

Solution: (b).

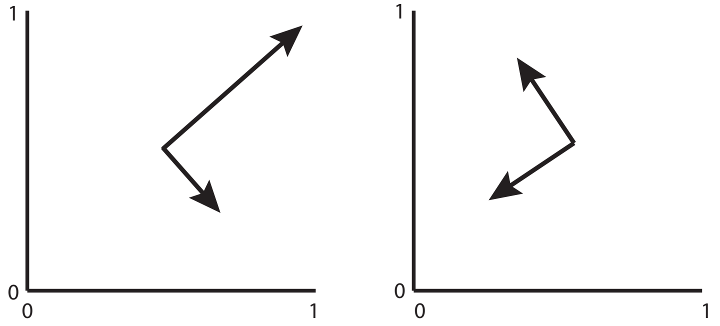
# 6 Principal Component Analysis (0 points)

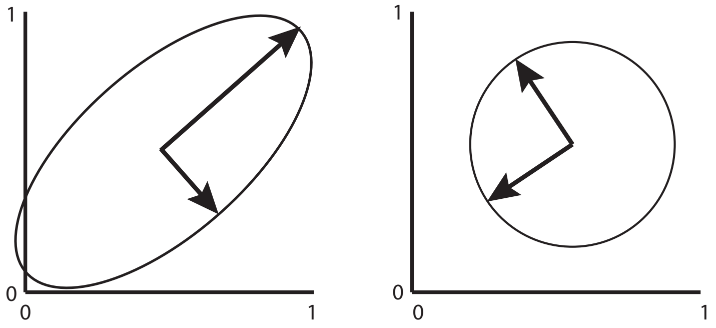
6.1. (a) **Drawing:** Consider the following two plots of data. Draw arrows from the mean of the data to denote the direction and relative magnitudes of the principal components.





(b) **Drawing:** 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.




1
0
0
1
1
0
0
1

6.2. Answer whether the following statements are true or false and if space is provided, provide a brief justification.

(a) **True or False:** 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

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). While performing PCA, the output variable is never provided.

(b) **True or False:** The output of PCA is a new representation of the data that is always of lower dimensionality than the original feature representation.

○ True

○ False

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

6.3. Given a dataset $\mathcal{D}$ consisting of $N$ data points and $D$ features, suppose you use PCA to project the dataset down to $d < D$ dimensions: let $E$ be the squared reconstruction error of this projection.

(a) **Select one:** Now suppose that you add an extra data point to $\mathcal{D}$ so that $\mathcal{D}'$ consists of $N + 1$ data points and $D$ features. You once again use PCA to project $\mathcal{D}'$ to $d < D$ dimensions: let $E'$ be the squared reconstruction error of this new projection. How do $E$ and $E'$ relate to one another?

○ $E < E'$

○ $E \leq E'$

○ $E = E'$

○ $E \geq E'$

○ $E > E'$

B

(b) **Select one:** Now suppose that you use PCA to project the original dataset $\mathcal{D}$ down to $(d+1) < D$ dimensions instead of $d$ dimensions: let $E'$ be the squared reconstruction error of this new projection. How do $E$ and $E'$ relate to one another?

○ $E < E'$

○ $E \leq E'$

○ $E = E'$

○ $E \geq E'$

○ $E > E'$

D

# 7 Reinforcement Learning (0 points)

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:

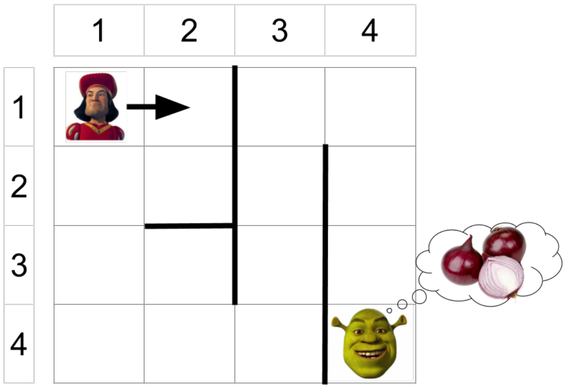


Figure 1: Kingdom of Duloc, circa 2001

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 be 1 if $s'$ is reachable from $s$ and by taking $a$, and 0 if not.

7.1. **Numerical answer:** What are $|S|$ and $|A|$ (size of state space and size of action space)?

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

7.2. **Short answer:** 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).

7.3. **Numerical answer:** What are the following transition probabilities?

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

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

Fix $\gamma = 0.5$ for following problems.

7.4. Given a start position of $(1,1,E)$, what is the expected discounted future reward from $a = R$? For $a = L$?

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:

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

7.5. **Short answer:** Farquaad's chief strategist (Vector from Despicable Me) suggests that using $\gamma = 0.9$ will result in a different optimal policy. 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!)

7.6. **Short answer:** 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)

7.7. 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...

7.8. **Numerical answer:** 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 total reward 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 $R_{exp} = 1 + 2 = 3$

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

☐ 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).

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

☐ 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.

7.11. **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 $R(s,a,s') \geq 0 \; \forall s,s' \in S, \; a \in A$.

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

○ $=$

○ $<$

○ $>$

○ $\leq$

○ $\geq$

○ ?

E

7.12. Answer whether the following statements are true or false and if space is provided, provide a brief justification.

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

○ True

○ False

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

(b) **True or False:** You can apply value or policy iteration to any problem that Q-learning can be applied to.

○ 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) **True or False:** Q-learning is always guaranteed to converge to the true value Q* for a greedy policy.

○ 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.

7.13. **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.

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

○ $=$

○ $<$

○ $>$

○ $\leq$

○ $\geq$

○ ?

F

7.14. **Select one:** During tabular Q-learning, you get a reward $r$ after taking action $North$ from state $A$ and arrive 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?

○ Q(A, North)

○ Q(A, South)

○ Q(B, North)

○ Q(B, South)

A

7.15. In general, for tabular Q-Learning to converge to the optimal Q function, which of the following are true?

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

○ True

○ False

**True or False:** It is necessary that the discount $\gamma$ is less than 0.5.

○ True

○ False

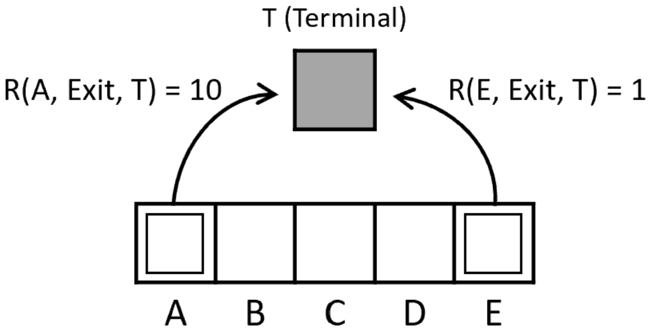
**True or False:** It is necessary that actions get chosen according to $\arg\max_a Q(s, a)$.

○ True

○ False

(1) **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. (2) **False**: The discount factor must be greater than 0 and less than 1, not 0.5. (3) **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.

7.16. 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.
- Valid actions in states B, C, and D are Left and Right.
- The only action from states A and E is Exit, which lead 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 and the second 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.

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

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

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

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

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 (0 points)

8.1. Random forests reduce the variance of single decision trees by introducing randomness at different stages of the algorithm: what are the places where we introduce this randomness? For each assertion below, indicate whether it is true or false and briefly justify your answer in 1-2 concise sentences.

(a) **True or False:** Bootstrap Aggregation - we choose $N$ random examples without replacement from the dataset every time we train a decision tree in the forest. Doing so ensures that every tree looks at a different set of examples and thus, the forest as a whole will not overfit to the dataset.

False, we choose them with replacement.

(b) **True or False:** Bagging - Take $N$ random examples with replacement from the dataset every time we train a decision tree in the forest. Doing so and then combining the hypothesis of all trees (by taking majority) reduces variance while still holding the trends and statistical properties of the original dataset.

True

(c) **True or False:** Feature Split Randomization - Every tree starts with a random subset of features and uses ID3 to split them. This ensures that all trees are not dependent on the same set of features and the forest is robust

False: Every node uses a random subset to make a spilt.

(d) **True or False:** Hypothesis Combination/Aggregation - We take the majority vote from a random subset of the decision trees at the end. This means that not all decision trees contribute to the final prediction, so the aggregated model is resilient to some trees having high variance.

False: We aggregate results from all the trees. We do not randomly choose N of them to make the final prediction.

8.2. 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 |

(a) **Numerical answer:** What is the OOB error of the above random forest classifier?

□ OOB is 3/7. Majority vote predictions for samples 2, 6, and 7 are incorrect.

(b) **Select one:** To reduce the error of each individual decision tree, Neural uses all the features to train each tree instead of using split-feature randomization. How would this impact the generalization error of the random forest?

○ The generalization error would decrease as each tree has lower generalization error

○ The generalization error would increase as each tree has insufficient training data

○ The generalization error would increase as the trees are highly correlated

The generalization error would increase as the trees are highly correlated

8.3. **Select all that apply:** In the AdaBoost algorithm, if the final hypothesis makes no mistakes on the training data, which of the following is correct?

- □ 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.
- □ None of the above

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

8.4. **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.

8.5. In the last semester, someone used AdaBoost on a dataset and recorded all the weights in each iteration 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 weighted training error less than $\frac{1}{2}$ was chosen at each step.

(a) **Numerical answer:** 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.

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

(b) **Short answer:** 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 determined, write 'Not Sure' instead.

| 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 | $\frac{1}{8}$ | $\frac{3}{8}$ | $\frac{1}{8}$ | $\frac{2}{8}$ | $\frac{3}{8}$ | $\frac{1}{8}$ |

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

Not sure

(c) **Select one:** You observe that the weights in round 3017 or 8888 (or both) cannot possibly be right. Which one is incorrect? Briefly explain your answer in 1-2 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.