Solutions

10-601 Machine Learning
Spring 2020
Exam 3 Practice Problems
May 2, 2020
Time Limit: N/A

Name:
Andrew Email:
Room:
Seat:
Exam Number:

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!

Topic	**Pages**	**# Questions**
Ensemble Methods	2	8
Recommender Systems	5	3
Hidden Markov Models	7	3
Graphical Models	10	2
Principal Component Analysis	13	2
Reinforcement Learning	15	6
K-Means	20	2
Support Vector Machines	24	8
Kernel Methods	27	7

1 Ensemble Methods

1. **[3pts]** 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.

2. **[3pts]** Which of the following is true about ensemble method?

   Select all that apply:

   □ Ensemble methods combine together many simple, poorly performing classifiers in order to produce a single, high quality classifier.

   □ Neural networks can be used in the ensemble methods.

   □ For the weighted majority algorithm, the weak classifiers are learned along the way.

   □ For the weighted majority algorithm, we want to give higher weights to better performing models.

   ABD

3. **[2pt] 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.

4. **[2pt] True or False:** AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

   ○ True

   ○ False

   False. Not if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using. For example consider the EXOR example with decision stumps as weak classifiers. No matter how many iterations are performed zero training error will not be achieved.

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

5. **[12pts]** 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.

   (a) **[3pts]** 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						

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

   (b) **[3pts]** 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

(c) [**3pts**] During Round 220, 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 221? List all the points that were misclassified. If none were misclassified, write 'None'. If it can't be decided, write 'Not Sure' instead.

ABE

(d) [**3pts**] 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.

NOTE: Please do not change the size of the following text box, and keep your answer in it. Thank you!

Your answer.

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

6. [**3 pts**] What condition must a weak learner satisfy in order for boosting to work?
Short answer:

The weak learner must classify above chance performance.

7. [**3 pts**] 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.

8. [**3 pts 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.

   Almost all coherent answers should be accepted, with full points to those who strongly argue their position with ML ideas.

2 Recommender Systems

1. [**4pts**] In which of the following situations will a collaborative filtering system be the most appropriate learning algorithm compared to linear or logistic regression?

   Select all that apply:

   □ You manage an online bookstore and you have the book ratings from many users. For each user, you want to recommend other books she will enjoy, based on her own ratings and the ratings of other users.

   □ You run an online news aggregator, and for every user, you know some subset of articles that the user likes and some different subset that the user dislikes. You'd want to use this to find other articles that the user likes.

   □ You've written a piece of software that has downloaded news articles from many news websites. In your system, you also keep track of which articles you personally like vs. dislike, and the system also stores away features of these articles (e.g., word counts, name of author). Using this information, you want to build a system to try to find additional new articles that you personally will like.

   □ You manage an online bookstore and you have the book ratings from many users. You want to learn to predict the expected sales volume (number of books sold) as a function of the average rating of a book.

   AB

2. [**3pts**] What is the basic intuition behind matrix factorization?

   Select all that apply:

   □ That content filtering and collaborative filtering are just two different factorizations of the same rating matrix.

   □ That factoring user and item matrices can partition the users and items into clusters that can be treated identically, reducing the complexity of making recommendations.

   □ The user-user and item-item correlations are more efficiently computed by factoring matrices.

   □ That user-item relations can be well described in a low dimensional space that can be computed from the rating matrices.

   D

3. **[3pts]** When building a recommender system using matrix factorization, the regularized objective function we wish to minimize is:

$$J(\mathbf{W},\mathbf{H}) = \sum_{u,i\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T\mathbf{h}_i)^2 + \lambda(\sum_u ||\mathbf{w}_u||^2 + \sum_i ||\mathbf{h}_i||^2)$$

where $\mathbf{w}_u$ is the uth row of $\mathbf{W}$ and the vector representing user u; $\mathbf{h}_i$ is the ith row of $\mathbf{H}$ and the vector representing item i; $\mathcal{Z}$ is the index set of observed user/item ratings in the training set; and λ is the weight of the L2 regularizer. One method of solving this optimization problem is to apply Block Coordinate Descent. The algorithms proceeds as shown below:

```
while not converged:
    for $u$ in $\{1, \ldots, N_u\}$:
        $\mathbf{w}_{u'} \leftarrow \arg\min_{\mathbf{w}_{u'}} J(\mathbf{W},\mathbf{H})$
    for $i$ in $\{1, \ldots N_i\}$
        $\mathbf{h}_{i'} \leftarrow \arg\min_{\mathbf{h}_{i'}} J(\mathbf{W},\mathbf{H})$
```

Doing so yields an algorithm called Alternating Least Squares (ALS) for matrix factorization. Which of the following is equal to the *transpose* of $\arg\min_{\mathbf{w}_{u'}} J(\mathbf{W},\mathbf{H})$?
Select one:

○ $v_u H(H^T H + \lambda I)^{-1}$

○ $(H^T H + \lambda I)^{-T} v_u H$

○ $v_u H(H^T H + \lambda I)^{-T}$

○ $v_u H(H^T H)^{-1}$

A or C

$$\frac{\partial L}{\partial \mathbf{w}_u} = -2\sum_i \left(v_{ui} - \mathbf{w}_u^\top \cdot \mathbf{h}_i\right)\mathbf{h}_i^\top + 2\lambda\mathbf{w}_u^\top \quad (1)$$

$$0 = -\left(\mathbf{v}_u - \mathbf{w}_u^\top H^\top\right)H + \lambda\mathbf{w}_u^\top \quad (2)$$

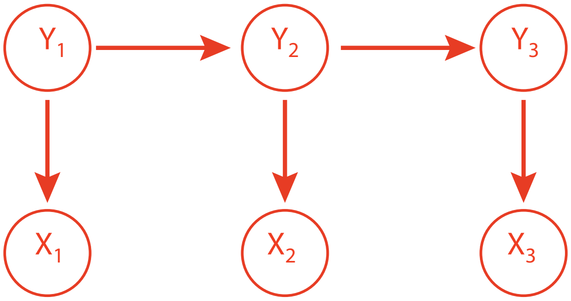
$$\mathbf{w}_u^\top\left(H^\top H + \lambda I\right) = \mathbf{v}_u H \quad (3)$$

$$\mathbf{w}_u^\top = \mathbf{v}_u H\left(H^\top H + \lambda I\right)^{-1} \quad (4)$$

3 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.

   (a) [**2 pts**] 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.



   (b) [**2 pts**] 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.
   $P(\mathbf{X}, \mathbf{Y}) =$

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

   (c) [**2 pts**] 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 $\boldsymbol{A} = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$, and emission matrix $\boldsymbol{B} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$.

   (a) [**3 pts**] What is $P(Y_5 = S_3)$?

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

(b) [**2 pts**] What is $P(Y_5 = S_3 | X_{1:7} = AABCABC)$?

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

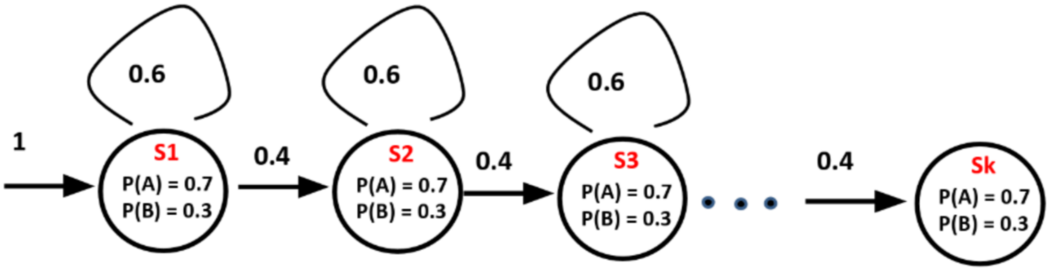
(c) [**4 pts**] Fill in the following table assuming the observation $AABCABC$. The α's are values obtained during the forward algorithm: $\alpha_t(i) = P(X_1, ..., X_t, Y_t = i)$.

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	$1/2^{13}$

(d) [**3 pts**] Write down the sequence of $Y_{1:7}$ with the maximal posterior probability assuming the observation $AABCABC$. What is that posterior probability?
$S_1S_1S_1S_2S_2S_3S_3$

posterior probability = 1



3. Consider the HMM in the figure above. 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. Transition probabilities for all states except s_k are also the same as shown. Once a run reaches s_k it outputs a symbol based on the s_k state emission probability and terminates.

   1. [**5 pts**] 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. [**9 pts**] 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

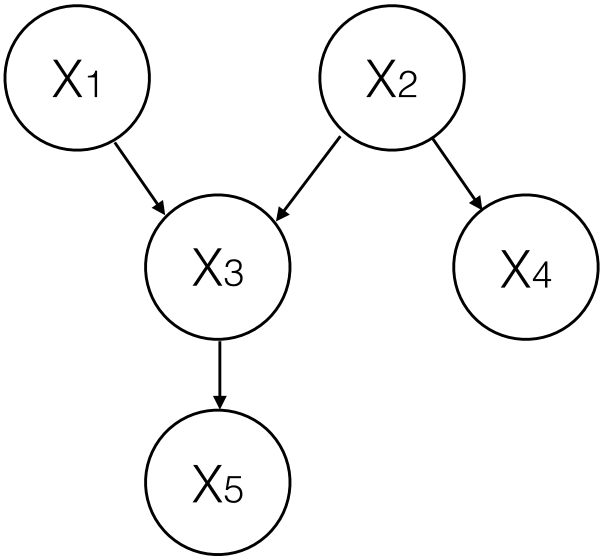
(c) $P('AAABA') \boxed{\qquad\qquad} P('BBAB')$

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

4 Graphical Models [16 pts]

1. Consider the following two Bayesian networks.

   (a) Answer whether the following conditional independence is true.



[2 pts] $X_1 \perp X_2 \mid X_3$?

Circle one: Yes No

Please explain briefly in one sentence.

No.

[2 pts] $X_1 \perp X_4$?

Circle one: Yes No

Please explain briefly in one sentence.

Yes.

[2 pts] $X_5 \perp X_2 \mid X_3$?

Circle one: Yes No

Please explain briefly in one sentence.

Yes.

(b) [4 pts] **Write out the joint probability in a form that utilizes as many independence/conditional independence assumptions contained in the graph as possible. Answer:** $P(X_1, X_2, X_3, X_4, X_5) =$

$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_3)$

(c) [**2 pts**] In the Hidden Markov Model (HMM), a state depends only on the corresponding observation and its previous state.

Circle one: True False

True.

(d) [**2 pts**] In a graphical model, if $X_1 \perp X_2$, then $X_1 \perp X_2|Y$ for every node Y in the graph.

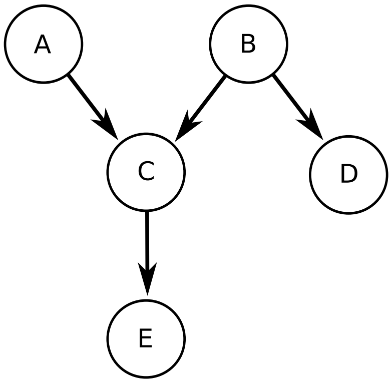
Circle one: True False

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

(e) [**2 pts**] In a graphical model, 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 graphical model shown above for questions (a)-(f). Assume all variables are boolean-valued.

(a) [2 pt.] (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.

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

(b) [2 pt.] **T or F**: Is C conditionally independent of D given B (i.e. is $(C \perp D)|B$)? Yes

(c) [2 pt.] **T or F**: Is A conditionally independent of D given C (i.e. is $(A \perp D)|C$)? No

(d) [2 pt.] **T or F**: Is A independent of B (i.e. is $A \perp B$)? Yes

(e) [4 pt.] 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.

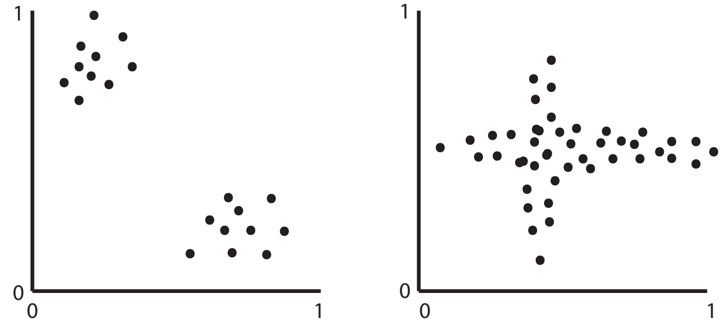
$$P(C = 1|A = 1, B = 0, D = 1, E = 0) = \frac{P(A = 1, B = 0, C = 1, D = 1, E = 0)}{\sum_{c=0}^{1} P(A = 1, B = 0, C = c, D = 1, E = 0)}$$

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

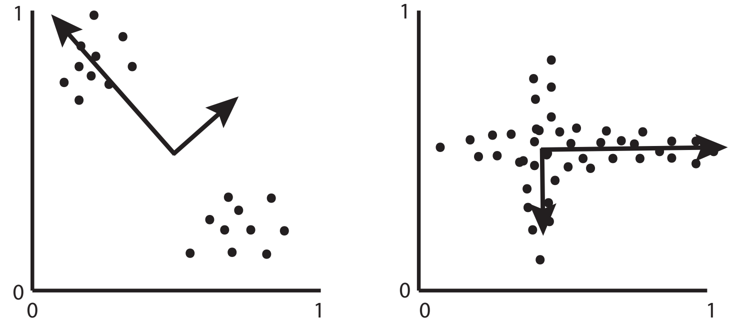
(f) [4 pt.] Draw a directed graphical model for the following situation: consider a sequence of random variables representing the total daily precipitation in the city of Pittsburgh, one for each day in a week. Additionally, for each day in the week, consider a random variable representing the total daily volume of snowfall on the 6th floor porch of the Gates building. We observe the snowfall on the porch of the Gates building, and would like to infer the total daily precipitation in Pittsburgh. The daily precipitation is dependent only on the the previous day's precipitation, and a given day's snowfall is dependent only on that day's precipitation. You may assume that the first day's total precipitation depends on nothing. Please make sure to label each node in your model with a variable name and then define all variables. Also make sure to indicate which variables are hidden and which are observed.

5 Principal Component Analysis

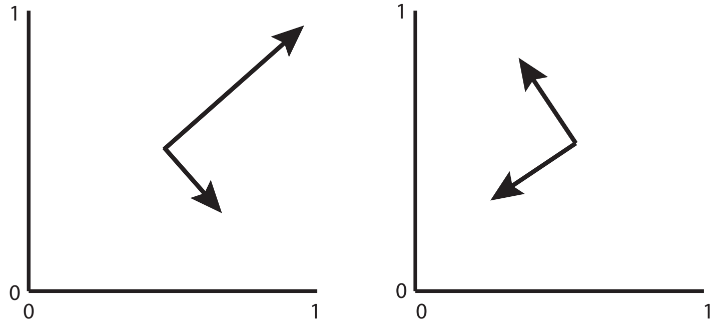
1. (i) **[5 pts]** Consider the following two plots of data. Draw arrows from the mean of the data to denote the direction and relative magnitudes of the principal components.



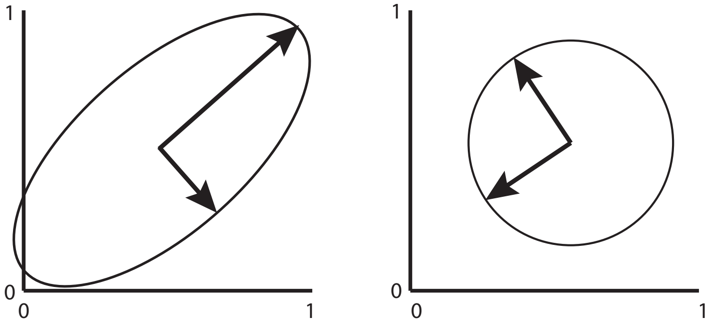
Solution:



(ii) **[5 pts]** 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 $Z^T U = \hat{X}$.

   (i) **[2 pts]** Removing the last row of U will still result in an approximation of X, but this will never be a better approximation than $\hat{X}$.

   Circle one: True False

   True.

   (ii) **[2 pts]** $\hat{X}\hat{X}^T = Z^T Z$.

   Circle one: True False

   True.

   (iii) **[2 pts]** 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.

   Circle one: True False

   False

   (iv) **[2 pts]** The output of PCA is a new representation of the data that is always of lower dimensionality than the original feature representation.

   Circle one: True False

   False

6 Reinforcement Learning

6.1 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:

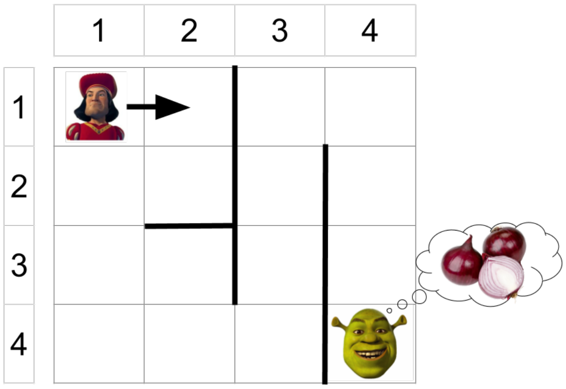


Figure 1: Kingdom of Duloc, circa 2001

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 × 4 columns × 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. Now 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)

6.2 Value and Policy Iteration

1. 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 $O(|A||S|^2)$

2. 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

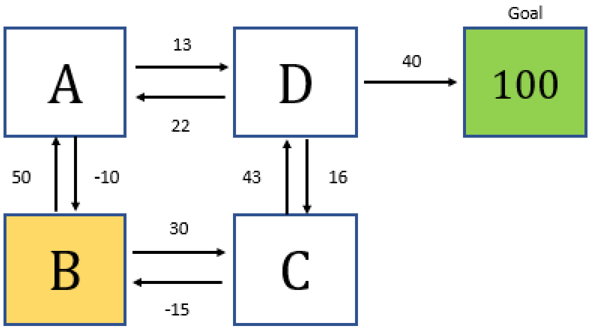
   □ Unknown Reward Function

   □ None of the Above

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

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. 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.



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

Up

(ii) 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)

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

C

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

174 (30 from the immediate action, and 144 from the future reward (value at state C))

6.3 Q-Learning

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

   (i) 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

(ii) 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.

(iii) 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:

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

(i) 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

(ii) Is this update rule synchronous or asynchronous?

Asynchronous

(iii) 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.

7 K-Means

1. For **True or False** questions, circle your answer and justify it; for **QA** questions, write down your answer.

   (i) For a particular dataset and a particular k, k-means 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.

   (ii) 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.

   (iii) The cluster assignments for all data points may not change at all between two consecutive iterations in k-means.

   ○ True

   ○ False

   Justify your answer:

   True. This will happen when k-means reaches the global or local optimum.

   (iv) 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.

(v) k in k-nearest neighbors and k-means has 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.

(vi) 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.

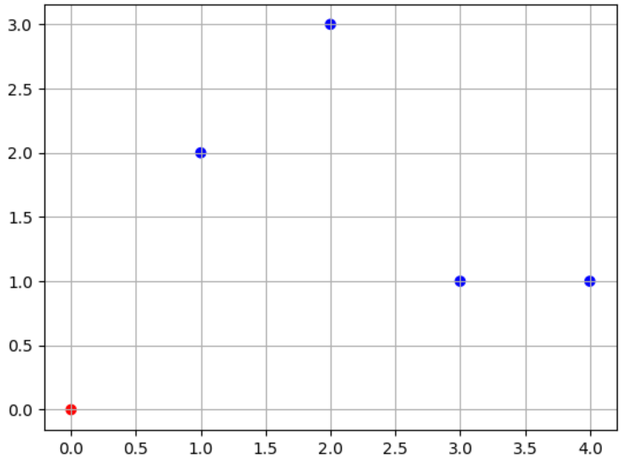
2. In k-means, random initialization could possibly lead to a local optimum with very bad performance. To alleviate this issue, instead of initializing all of the centers completely randomly, we decide to use a smarter initialization method. This leads us to k-means++.

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 choosen as (0, 0). These data points are shown in the figure below.



(i) [**5 pts**] 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): 0.170
(2, 3): 0.275
(3, 1): 0.241
(4, 1): 0.314

(ii) [**1 pts**] Which data point is mostly liken chosen as the center for cluster 2?

(4, 1) is mostly likely chosen.

(iii) [**5 pts**] 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): 0.369
(2, 3): 0.466
(3, 1): 0.165
(4, 1): 0

(iv) [**1 pts**] Which data point is mostly liken chosen as the center for cluster 3?

(2, 3) is mostly likely chosen.

(v) [**3 pts**] 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)

(vi) **[3 pts]** 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)

(vii) **[2 pts]** 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 clusters, to make sure all of the initial cluster centers stay away from each other.

8 Support Vector Machines

1. **[3 pts.]** Given the same training data, in which the points are linearly separable, the margin of the decision boundary produced by SVM will always be greater than or equal to the margin of the decision boundary produced by Perceptron.

   Circle one: True False

   True. Because svm is managed to solve the classification problem with largest possible margin.

2. **[2 pts]** The support vectors for a soft margin SVM include the points within the margin as well as those that are incorrectly classified.

   Circle one: True False
   One line justification (only if False): True.

3. **[2 pts]** If the data is linearly separable, SVM minimizes $\|w\|^2$ subject to the constraints $\forall i, y_i w \cdot x_i \geq 1$. In the linearly separable case, which of the following may happen to the decision boundary if one of the training samples is removed? **Circle all that apply.**

   - Shifts toward the point removed
   - Shifts away from the point removed
   - Does not change

   AC

4. **[3 pts]** Recall that when the data are not linearly separable, SVM minimizes $\|w\|^2 + C\sum_i \xi_i$ subject to the constraint that $\forall i$, $y_i w \cdot x_i \geq 1 - \xi_i$ and $\xi_i \geq 0$. The tradeoff parameter C allows for errors in the training samples. Which of the following may happen to the size of the margin if the tradeoff parameter C is increased? **Circle all that apply.**

   - Remains the same
   - Increases
   - Decreases

   Remains the same or decreases. If increase C doesn't affect the support vector, then the size of the margin will remain the same. If increase C reduces some support vector, then the margin becomes smaller.

5. SVM is a discriminative classifier, whereas Naïve Bayes is a generative classifier. Describe one statistical advantage SVM has over Naïve Bayes.
   Describe in one sentence:

   SVM does not need the independent assumption made in naive bayes.

6. [4 pt.] SVM and Perceptron both give linear decision boundaries. Which of the following are correct about the differences between SVM and perceptron?

   (a) SVM maximizes the margin of the classifier while perceptron does not necessarily

   (b) SVM can take advantage of the kernel trick while perceptron cannot

   (c) SVM can allow classification errors with soft margin while perceptron cannot

   (d) Perceptron is more suitable for online learning while SVM is less suitable

   (e) Perceptron has fewer parameters to learn while SVM has more

   (f) Perceptron is guaranteed to converge in the linearly separable case while SVM is not

   (a)(c)(d). (b): perceptron can also be kernelized, see question 6 in the kernel methods section. (e): the number of parameters for perceptron equals the number of features, for SVM equals the number of features (primal) or the number of samples (dual). Not necessarily one has more parameters than the other. (f): both are guaranteed to converge in the linearly separable case.

7. [4 pts.] Consider the dataset in Fig. 2. Under the SVM formulation in problem (3),

   (a) Draw the decision boundary on the graph.

   (b) What is the size of the margin?

   (c) Circle all the support vectors on the graph.

   $x_2 - 2.5 = 0$. The size of margin is 0.5. Support vectors are x_2, x_3, x_6, x_7.

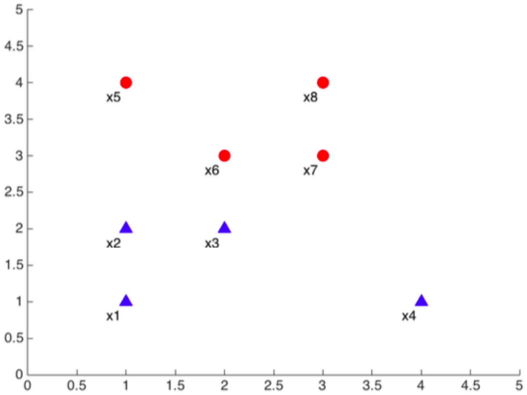


Figure 2: SVM toy dataset

8. **[Extra Credit: 3 pts.]** One formulation of soft-margin SVM optimization problem

is:

$$
\begin{aligned}
\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\
\text{s.t. } y_i(\mathbf{w}^\top x_i) \geq 1 - \xi_i \quad \forall i = 1, ..., N \\
\xi_i \geq 0 \quad \forall i = 1, ..., N \\
C \geq 0
\end{aligned}
$$

where (x_i, y_i) are training samples and $\mathbf{w}$ defines a linear decision boundary.

Derive a formula for ξ_i when the objective function achieves its minimum (No steps necessary). Note it is a function of $y_i\mathbf{w}^\top x_i$. Sketch a plot of ξ_i with $y_i\mathbf{w}^\top x_i$ on the x-axis and value of ξ_i on the y-axis. What is the name of this function?

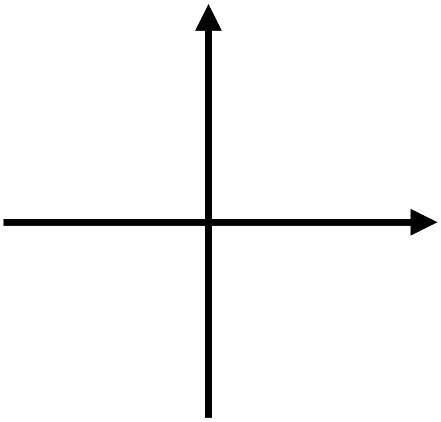


Figure 3: Plot here

Hinge Loss.

9 Kernel Methods

1. **[3 pts.]** Applying the kernel trick enables features to be mapped into a higher dimensional space, at a cost of higher computational complexity to operate in the higher dimensional space.

   Circle one: True False

   False. We didn't increase computational complexity, that's the whole point of kernel trick.

2. **[3 pts.]** Since the VC dimension for an SVM with a Radial Base Kernel is infinite, such an SVM must have a larger generalization error than an SVM without kernel which has a finite VC dimension.

   Circle one: True False

   False. The learning theory gives a *trivial* upper bound for generalization error when VC dimension is infinite, and it doesn't imply that RBF kernel won't work in practice.

3. **[3 pts.]** Suppose $\phi(x)$ is an arbitrary feature mapping from input $x \in \mathcal{X}$ to $\phi(x) \in \mathbb{R}^N$ and let $K(x, z) = \phi(x) \cdot \phi(z)$. Then $K(x, z)$ will always be a valid kernel function.

   Circle one: True False

   True. This is the definition of a kernel function.

4. **[3 pts.]** Suppose $\phi(x)$ is the feature map induced by a polynomial kernel $K(x, z)$ of degree d, then $\phi(x)$ should be a d-dimensional vector.

   Circle one: True False

   False. The dimension of $\phi(x)$ is not a constant but increases with the dimension of x. For example if $x \in \mathbb{R}^2$ and $K(x, z) = (1 + x^T z)^2$ has a degree of $d = 2$, then $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2)^T$.

5. **[3 pts.]** The decision boundary that we get from a Gaussian Naive Bayes model with class-conditional variance is quadratic. Can we in principle reproduce this with an SVM and a polynomial kernel?

   Circle one: Yes No

   Yes. Quadratic decision boundaries can be reproduced with an SVM with polynomial kernel of degree two.

6. Let's go kernelized!

   (a) **Perceptron review.** Assume we have a binary classification task with dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{\infty}$ where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{-1, 1\}$. Recall that the perceptron learns a linear classifier $y = sign(w^T x)$ by applying the following algorithm.

```
Algorithm 1: Perceptron algorithm
Initialize the weights $w = 0$;
for $i = 1, 2, \cdots$ do
    Predict $\hat{y}^{(i)} = sign(w^T x^{(i)})$;
    if $\hat{y}^{(i)} \neq y^{(i)}$ then
        Update $w = w + y^{(i)} x^{(i)}$;
end
Final classifier: $h(x) = sign(w^T x)$
```

Show that the final weight vector w is a linear combination of all the samples $x^{(i)}$ $(i = 1, 2, \cdots, T)$ it has been trained on, and hence for prediction we can write $w^T x$ in the form of $w^T x = \sum_{i=1}^T \alpha_i K(x^{(i)}, x)$ for some α_i where $K(x, z) = x^T z$.

The final weight can be written as $w = \alpha_1 x^{(1)} + \cdots + \alpha_T x^{(T)}$ where $\alpha_i = y^{(i)}$ if a mistake is made at iteration i and 0 otherwise. Hence $w^T x = \sum_{i=1}^T \alpha_i x^{(i)T} x = \sum_{i=1}^T \alpha_i K(x^{(i)}, x)$.

(b) **Kernelized perceptron.** Now we are going to introduce a kernel function $K(x, z)$ to kernelize the perceptron algorithm. Based on your findings in the previous question, fill in the blanks below to complete the kernelized perceptron algorithm using the kernel $K(x, z)$. Assume the training loop stops after it has seen T training samples.

```
Algorithm 2: Kernelized Perceptron
Initialize ____________________________;
for $i = 1, 2, \cdots$ do
    Predict $\hat{y}^{(i)}$ = ____________________________;
    if $\hat{y}^{(i)} \neq y^{(i)}$ then
        ____________________________;
end
Final classifier: $h(x)$ = ____________________________.
```

(1) Initialize all α_i to 0;

(2) $\hat{y}^{(i)} = sign(\sum_{j=1}^{i-1} \alpha_j K(x^{(j)}, x))$;

(3) Update $\alpha_i = \alpha_i + y^{(i)}$ (or Set $\alpha_i = y^{(i)}$);

(4) $h(x) = sign(\sum_{j=1}^{T} \alpha_j K(x^{(j)}, x))$;

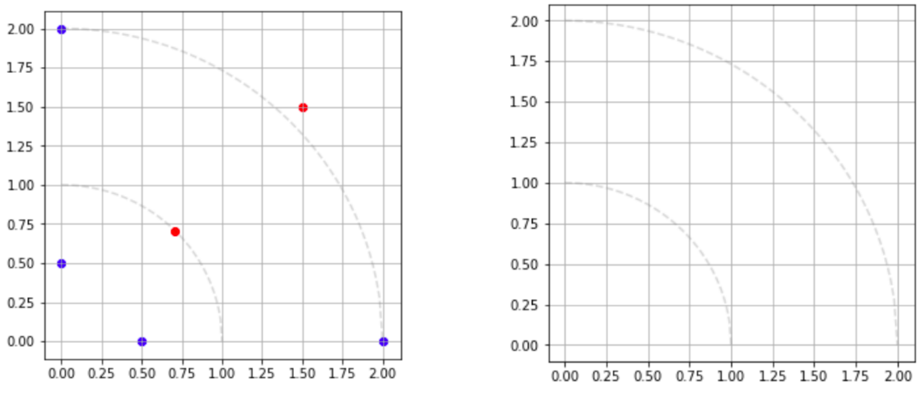
(c) **Short answer.** Describe one advantage and one disadvantage of using kernelized perceptron compared to using vanilla perceptron.

Advantage example: kernels can introduce complex features to perceptron, so that non-linear decision boundaries can be learned and the resulting model should be more powerful.

Disadvantage example: kernelized perceptron has to store all training samples it has seen. This requires a lot of storage and yields a higher time complexity for prediction when the number of training samples is large.

7. Suppose we have six training samples that lie in a two-dimensional space as is shown in Figure 4a. Four of them belong to the blue class: $(0, 0.5), (0, 2), (0.5, 0), (2, 0)$, and two of them belong to the red class: $(\sqrt{2}/2, \sqrt{2}/2), (1.5, 1.5)$. Unfortunately, this dataset is not linearly separable. You recall that kernel trick is one technique you can take advantage of to address this problem. The trick uses a kernel function $K(x, z)$ which implicitly defines a feature map $\phi(x)$ from the original space to the feature space. Consider the following normalized kernel:

$$K(x, z) = \frac{x^T z}{\|x\|_2 \|z\|_2}.$$



(a) Data points in the original space

(b) Data points in the feature space

(a) What is the feature map $\phi(x)$ that corresponds to this kernel? Draw $\phi(x)$ for each training sample in Figure 4b.

$\phi(x) = x/\|x\|_2$. Blue points are mapped to (0, 1) and (1, 0). Red points are mapped to $(\sqrt{2}/2, \sqrt{2}/2)$.

(b) The samples should now be linearly separable in the feature space. The classifier in the feature space that gives the maximum margin can be represented as a line $w^T x + \alpha = 0$. Draw the decision boundary of this classifier in Figure 4b. What are the coefficients in the weight vector $w = (w_1, w_2)^T$? Hint: you don't need to compute them.

$w = (1, 1)^T$ due to symmetry. Observe that $1 \leq -\alpha \leq \sqrt{2}$, so the exact value of α if anyone is interested should be $-(1 + \sqrt{2})/2$.

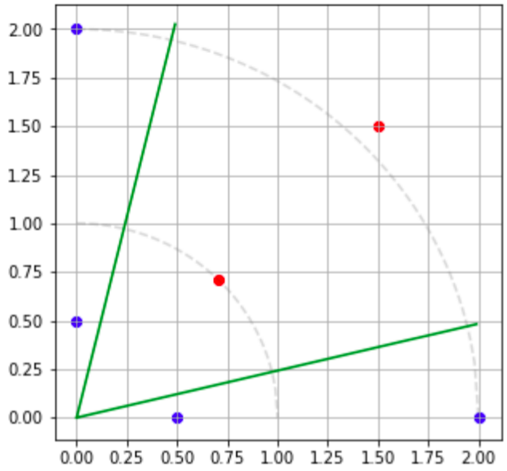
(c) Now we map the decision boundary obtained in (b) back to the original space. Write down the corresponding boundary in the original space in the format of an explicit equation. You can keep α in your equation. Try to plot its rough shape in Figure 4a.

The decision boundary in the original space is

$$w^T\phi(x)+\alpha=0 \implies \frac{x_1+x_2}{\sqrt{x_1^2+x_2^2}}+\alpha=0 \implies (x_1+x_2)^2=\alpha^2(x_1^2+x_2^2)$$

$$\implies x_1^2+x_2^2-\frac{2}{\alpha^2-1}x_1x_2=0 \implies x_2=\frac{\eta\pm\sqrt{\eta^2-4}}{2}x_1 \;(\eta=\frac{2}{\alpha^2-1}>2)$$

The final step is obtained by solving the quadratic equation. It's okay if you didn't work to the simplest form. So the decision boundary would be two straight lines. A possible rough shape is shown below.


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00