CS273A Final Exam

Introduction to Machine Learning: Winter 2019

Thursday, March 21st, 2019

Your name: **Row/Seat Number:**

Your ID #(e.g., 123456789) **UCINetID (e.g.ucinetid@uci.edu)**

- Please put your name and ID **on every page**.
- Total time is 1 hour 50 minutes. READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Please **write clearly** and **show all your work**.
- If you need clarification on a problem, please raise your hand and wait for the instructor or TA to come over.
- You may use **one** sheet containing handwritten notes for reference, and a **basic** calculator; no other electronics allowed.

Problems

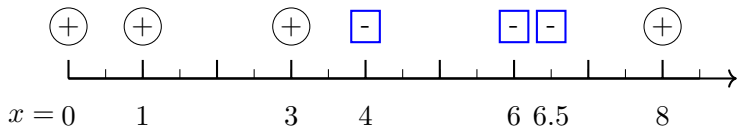
Total, *(79 points.)*

This page is intentionally blank, use as you wish.

Name: ID#:

K-Nearest Neighbors, *(8 points.)*

Consider the following dataset with *five* points shown below, for a binary classification task ($y = +, -$) with a scalar feature x. In case of ties, **prefer the negative class**. Put final answers in the box.



(1) Compute the **training** error of a 1-nearest neighbor classifier. *(2 points.)*

(2) Compute the **leave-one-out** cross-validation error of 1-nearest neighbor classifier. *(2 points.)*

(3) Compute the **training** error of 2-nearest neighbor classifier. *(2 points.)*

(4) Compute the **leave-one-out** cross-validation error of 2-nearest neighbor classifier. *(2 points.)*

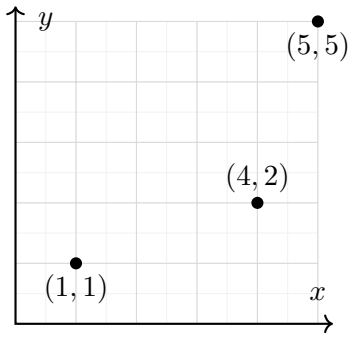
This page is intentionally blank, use as you wish.

Name: ID#:

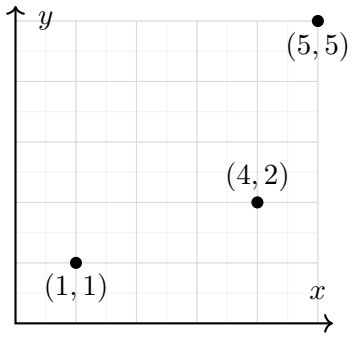
Linear Regression, *(8 points.)*

Consider the following data points, copied in each part. We wish to perform linear regression to minimize the mean squared error of our predictions.

(a) Compute the leave-one-out cross-validation error of a zero-order (constant) predictor, $\hat{y}(x) = \theta_0$. *(4 points.)*



(b) Compute the leave-one-out cross-validation error of a first-order (linear) predictor, $\hat{y}(x) = \theta_0 + \theta_1 x$. *(4 points.)*



This page is intentionally blank, use as you wish.

Name: _____ ID#: _____

Multiple Choice, *(14 points.)*

For each of the following statements, choose whether the statement is true or false.

Statement	True	False
Universal function approximators		
With enough hidden nodes and layers, a neural network can approximate any function.	☐	☐
With enough hidden nodes and a *single* layer, a neural network can approximate any function.	☐	☐
With enough data, a Gaussian Bayes classifier can approximate any function.	☐	☐
With enough depth, a decision tree can approximate any function.	☐	☐
Bagging can be used to make any simpler classifier arbitrarily complex.	☐	☐
Optimization		
Using backpropagation to train a neural network will avoid getting stuck in local optima.	☐	☐
Stochastic gradient descent is often preferred over batch when the number of data points m is very large.	☐	☐
Linear regression can be solved using either matrix algebra or gradient descent.	☐	☐
The K-Means algorithm is guaranteed to converge in finite number of steps.	☐	☐
The number of steps for agglomerative clustering to complete does not depend on the linkage function.	☐	☐

Order the following four learners in order of *increasing* complexity (likelihood of overfitting):

(a) Perceptron, $\hat{y} = T(\theta \cdot x)$

(b) Ensemble (mixture) of three perceptrons,

$$\hat{y} = \alpha_1 T(\theta_1 \cdot x) + \alpha_2 T(\theta_2 \cdot x) + \alpha_3 T(\theta_3 \cdot x)$$

trained by jointly optimizing $\{\alpha_1, \alpha_2, \alpha_3, \theta_1, \theta_2, \theta_3\}$

(c) Ensemble of three perceptrons trained by AdaBoost

(d) Ensemble of three perceptrons trained by Bootstrap Aggregation

(Simplest) ☐ ☐ ☐ ☐ (Most Complex)

This page is intentionally blank, use as you wish.

Name: _____ ID#: _____

Decision Trees, *(10 points.)*

Consider the table of measured data given at right. We will use a decision tree to predict the outcome y using the three features, $x_1, \ldots, x_3$. In the case of ties, we prefer to use the feature with the smaller index (x_1 over x_2, etc.) and prefer to predict class 1 over class 0. You may find the following values useful (although you may also leave logs unexpanded):

$\log_2(1) = 0 \quad \log_2(2) = 1 \quad \log_2(3) = 1.59 \quad \log_2(4) = 2$

$\log_2(5) = 2.32 \quad \log_2(6) = 2.59 \quad \log_2(7) = 2.81 \quad \log_2(8) = 3$

x_1	x_2	x_3	y
0	0	0	1
1	0	1	1
1	0	1	1
1	1	1	0
0	1	0	0
1	0	0	0

(1) What is the entropy of y? *(2 points.)*

(2) Which variable would you split first? Justify your answer. *(2 points.)*

(3) What is the information gain of the variable you selected in part (2)? *(3 points.)*

(4) Draw the rest of the decision tree learned on these data. *(3 points.)*

This page is intentionally blank, use as you wish.

Name: ID#:

Bayes Classifiers, *(10 points.)*

x_1	x_2	x_3	y
0	0	1	0
1	1	1	0
1	0	0	0
1	0	0	0
0	0	0	1
0	1	1	1
0	1	1	1
1	1	0	1

Consider the table of measured data given at right. We will use the three observed features x_1 , x_2 , x_3 to predict the class y. In the case of a tie, we will prefer to predict class $y = 0$.

(1) Write down the probabilities used by a naïve Bayes classifier: *(4 points.)*

$p(y = 0)$: $p(y = 1)$:

$p(x_1 = 1|y = 0)$: $p(x_1 = 1|y = 1)$:

$p(x_1 = 0|y = 0)$: $p(x_1 = 0|y = 1)$:

$p(x_2 = 1|y = 0)$: $p(x_2 = 1|y = 1)$:

$p(x_2 = 0|y = 0)$: $p(x_2 = 0|y = 1)$:

$p(x_3 = 1|y = 0)$: $p(x_3 = 1|y = 1)$:

$p(x_3 = 0|y = 0)$: $p(x_3 = 0|y = 1)$:

(2) Using your naïve Bayes model, compute: *(3 points.)*
$p(y = 1|x_1 = 0, x_2 = 1, x_3 = 1)$: $p(y = 0|x_1 = 0, x_2 = 1, x_3 = 1)$:

(3) Compute the probabilities $p(y = 1|x_1 = 0, x_2 = 1, x_3 = 1)$ and $p(y = 0|x_1 = 0, x_2 = 1, x_3 = 1)$ for a joint Bayes model trained on the same data. *(3 points.)*

This page is intentionally blank, use as you wish.

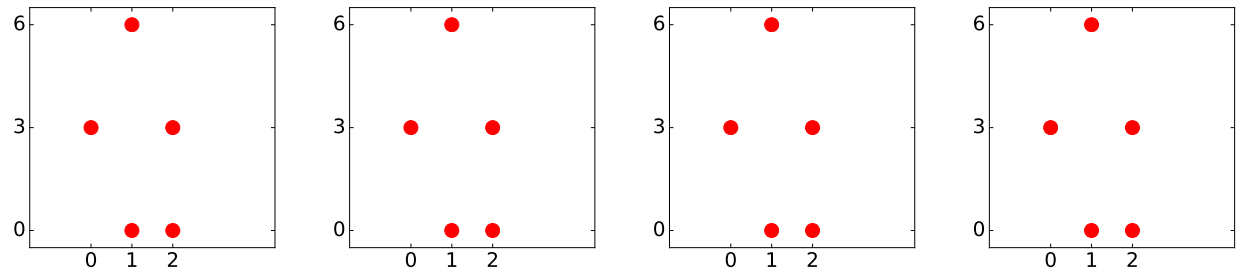
Name: ID#:

Clustering, *(10 points.)*

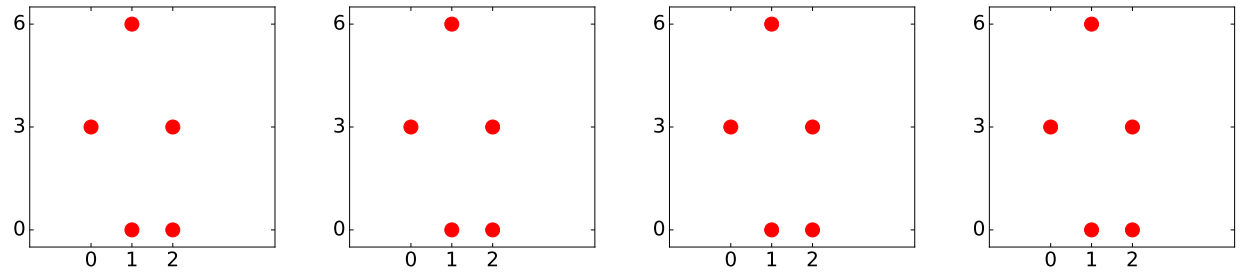
Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data.

Linkage

(a) Execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "single linkage" (minimum distance) for the cluster scores. Stop when the algorithm would terminate, or after 4 steps, whichever is first. Show each step separately in a panel.

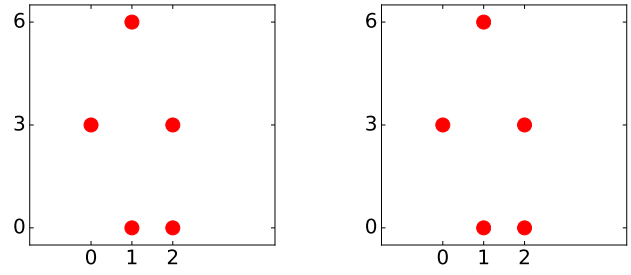


(b) Now repeat your agglomerative clustering algorithm, this time using "complete linkage" (maximum distance) for the cluster scores. Stop when the algorithm would terminate, or after 4 steps, whichever is first. Show each step separately in a panel.



k-means

In the panels below, sketch two different clustering solutions that could be found by (would be converged solutions of) k-means, with $k = 2$. Show the final (converged) grouping of data (assignments) and final cluster locations. (You do not have to show the process, only the final clustering.) This illustrates the existence of local optima in k-means.

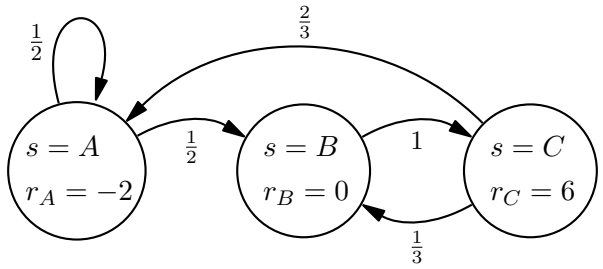


This page is intentionally blank, use as you wish.

Name: ID#:

Markov Processes, *(9 points.)*

Consider the Markov model shown here:



$\Pr[A \to A] = 0.5$
$\Pr[A \to B] = 0.5$
$\Pr[B \to C] = 1.0$
$\Pr[C \to B] = 0.33$
$\Pr[C \to A] = 0.66$

where the transition probabilities are shown next to each arc and at right, and the rewards r_s associated with each state s are shown inside the circles. Assume a future discounting factor of $\gamma = \frac{1}{2}$.

(1) Compute $J^1(s)$, the expected discounted sum of rewards for state sequences of length 1 (e.g., $[A]$) starting in each state s. *(3 points.)*

(2) Compute $J^2(s)$, the expected discounted sum of rewards for state sequences of length 2 (e.g., $[C \to B]$) starting in each state s. *(3 points.)*

(3) Compute $J^3(s)$, the expected discounted sum of rewards for state sequences of length 3 (e.g., $[B \to A \to C]$) starting in each state s. *(3 points.)*

This page is intentionally blank, use as you wish.

Name: _____ ID#: _____

VC-Dimension, *(10 points.)*

Consider a family of classifiers on two-dimensional data $x = (x_1, x_2)$ that use the *angle* of the vector x at the origin. We classify a point as positive ($\hat{y} = +1$) if it lies between two angles θ_a and θ_b (parameters of the classifier), moving counterclockwise.

More precisely, we have

$$\hat{y}(x\,;\,\theta_a, \theta_b) = \begin{cases} +1 & \theta_a \leq \arctan(\frac{x_2}{x_1}) + 2k\pi \leq \theta_b \qquad \text{for some } k \in \mathbb{Z} \\ -1 & \text{otherwise} \end{cases}$$

where θ_a, θ_b are parameters to be learned from the data.

(1) What is the the VC dimension H of this classifier? *(2 points.)*

(2) Demonstrate that H is at least the value you gave in Part (a). *(4 points.)*

(3) Show by counterexample that H is no larger than the value you gave in Part (a). *(4 points.)*

This page is intentionally blank, use as you wish.

This page is intentionally blank, use as you wish.

This page is intentionally blank, use as you wish.