CS 178 Final Exam

Machine Learning and Data Mining: Winter 2018

Wednesday, March 21, 2018 from 8:00-10:00am

Your Name: THE SOLUTION

Row/Seat Number:

Your ID #(e.g., 123456789)

UCINetID (e.g., ucinetid@uci.edu)

- Total time is **120 minutes**. Please look through all questions and organize your time wisely.
- Please put your name and ID **on every page**.
- Please write your **row and seat number** on the cover page.
- For full credit, be sure to **write clearly** and **show all your work**.
- **Electronic devices are not allowed.**
- You may use **one** (two-sided) 8.5x11-inch sheet of (your own) handwritten notes.
- Have a question? **Please raise your hand for help.**

Problems

1 **Binary Classification and Separability** *(8 points)* **3**

2 **Decision Trees** *(10 points)* **5**

3 **Cross-Validation for Nearest Neighbor Classification** *(9 points)* **7**

4 **Regression via Gradient Descent** *(11 points)* **9**

5 **VC Dimension** *(10 points)* **11**

6 **Clustering Algorithms** *(12 points)* **13**

7 **Multiple Choice** *(10 points)* **15**

Total *(70 points)*

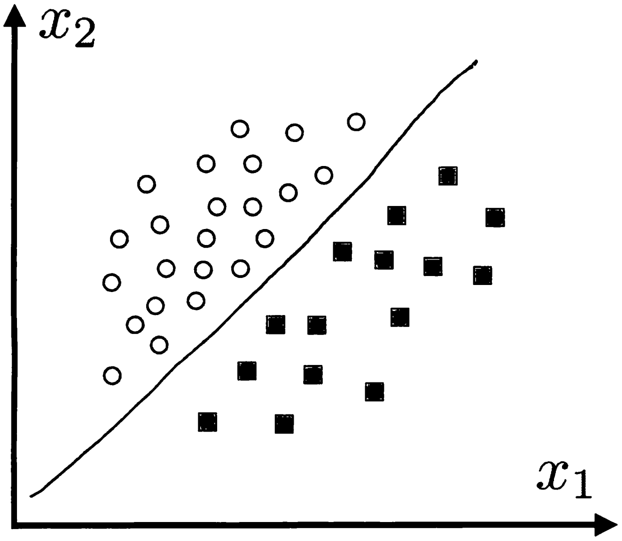
This page is intentionally blank, use as you wish.

Name: THE SOLUTION ID#:

Problem 1 Binary Classification and Separability *(8 points)*

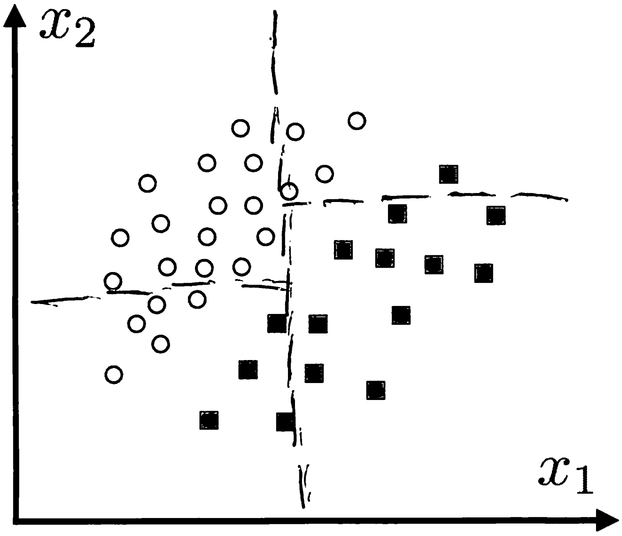
For the training datasets below, two input features x_1, x_2 will be used to predict a binary class variable y. Open circles indicate training examples of class $y = +1$, and filled squares indicate training examples of class $y = -1$.

(1) Can the data below be separated by a **perceptron classifier with linear features**? If yes, draw an example of a separating decision boundary. If no, briefly explain why.



YES

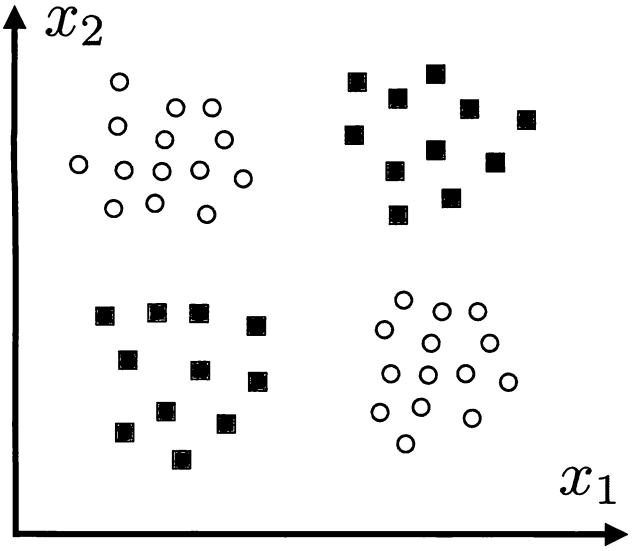
(2) Can the data below be separated by a **two-level decision tree**? If yes, draw an example of a separating decision boundary. If no, briefly explain why.



No

Two axis-parallel splits cannot separate all data.

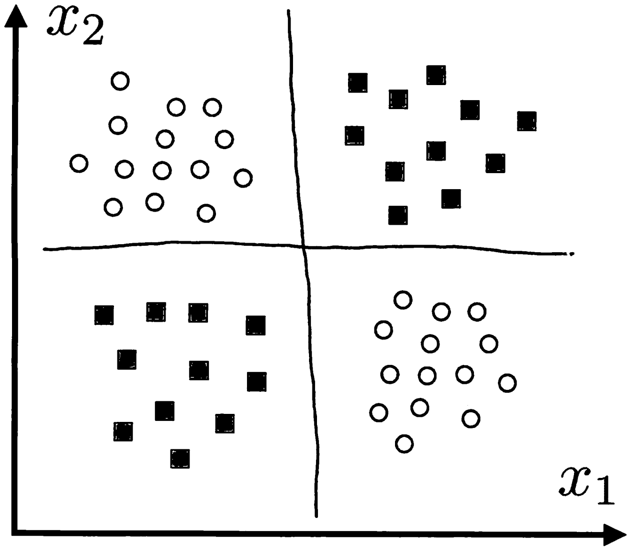
(3) Can the data below be separated by a **perceptron classifier with linear features**? If yes, draw an example of a separating decision boundary. If no, briefly explain why.



No

No linear decision boundary will separate this "XOR" pattern.

(4) Can the data below be separated by a **two-level decision tree**? If yes, draw an example of a separating decision boundary. If no, briefly explain why.



YES

Name: THE SOLUTION ID#:

Problem 2 Decision Trees *(10 points)*

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

Consider the table of training data given at right. We will use the three observed features x_1, x_2, x_3 to predict the binary class y. In the case of a tie, we will prefer to predict class $y = 1$. Note that some data may be repeated (observed more than once). When evaluating entropies, you may find the following constants useful (but you may also leave logs unexpanded):
$\log_2(1) = 0$, $\log_2(2) = 1$, $\log_2(3) = 1.59$, $\log_2(4) = 2$,
$\log_2(5) = 2.32$, $\log_2(6) = 2.59$, $\log_2(7) = 2.81$, $\log_2(8) = 3$.

(1) What is the entropy of the binary variable y?

Entropy: $H(p) = -p \log_2(p) - (1-p)\log_2(1-p)$

$$H\left(\frac{3}{6}\right) = H\left(\frac{1}{2}\right) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$$

$$= \log_2(2) = \boxed{1 \text{ bit}}$$

(2) Which feature would a decision tree learning algorithm choose to split first, in order to maximize the information gain? (You do *not* need to compute precise numerical values for the information gains, you only need to find the feature with the largest information gain.)

$$IG(x_i) = H(y) - P(X_i=0)H(Y|X_i=0) - P(X_i=1)H(Y|X_i=1)$$

$$IG(x_1) = H\left(\frac{1}{2}\right) - \frac{1}{3}H\left(\frac{1}{2}\right) - \frac{2}{3}H\left(\frac{1}{2}\right) = 0$$

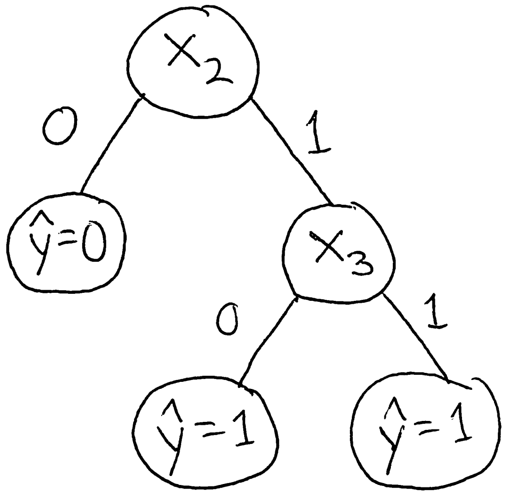
$$IG(x_2) = H\left(\frac{1}{2}\right) - \frac{1}{3}H(0) - \frac{2}{3}H\left(\frac{3}{4}\right) = 1 - \frac{2}{3}H\left(\frac{1}{4}\right)$$

$$IG(x_3) = H\left(\frac{1}{2}\right) - \frac{1}{2}H\left(\frac{2}{3}\right) - \frac{1}{2}H\left(\frac{1}{3}\right) = 1 - H\left(\frac{1}{3}\right)$$

We know that $0 < H\left(\frac{1}{4}\right) < H\left(\frac{1}{3}\right) < 1$, so

$$\boxed{X_2 \text{ maximizes information gain}}$$

(3) Suppose your decision tree learning algorithm is constrained to a maximum depth of 2, where the split for the first level is determined as in part (b). What decision tree will be learned? What is the training error rate of that decision tree?



For all data where $x_2=0$, $y=0$.

Data where $x_2=1$:

X_1	X_3	Y
1	1	0
1	0	1
1	0	1
0	1	1

$$IG(x_1) = H(\tfrac{1}{4}) - \tfrac{3}{4}H(\tfrac{1}{3}) - \tfrac{1}{4}H(1) = H(\tfrac{1}{4}) - \tfrac{3}{4}H(\tfrac{1}{3})$$

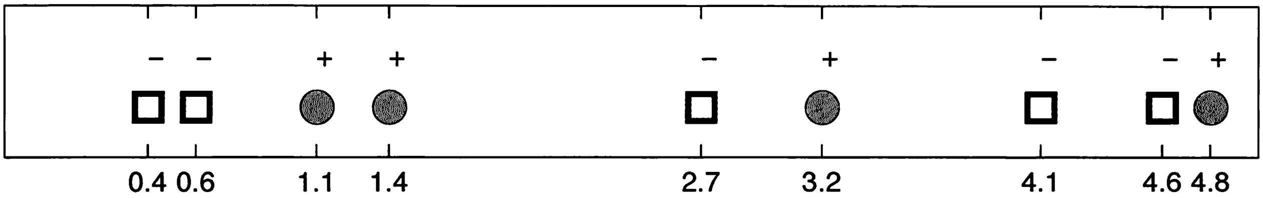
$$IG(x_3) = H(\tfrac{1}{4}) - \tfrac{1}{2}H(\tfrac{1}{2}) - \tfrac{1}{2}H(1) = H(\tfrac{1}{4}) - \tfrac{1}{2}$$

- Splitting on x_3 gives best informatn gain, which gives tree above with train error rate $\boxed{\frac{1}{6}}$

- But if split on x_1 instead, get alternatve 2-level tree where error rate is also $1/6$

Name: THE SOLUTION ID#:

Problem 3 Cross-Validation for Nearest Neighbor Classification *(9 points)*



Consider the training data above, where one scalar feature x (whose numerical values are below each data point) will be used to predict a binary class variable y. Filled circles indicate training examples of class $y = +1$, and open squares indicate training examples of class $y = -1$. We use a K-nearest-neighbor classifier, and in the case of ties, prefer to predict class -1.

(1) Compute the *leave-one-out cross-validation* error rate of a 1-Nearest-Neighbor classifier.

Points where errors made: 2.7, 3.2, 4.6, 4.8 (where closest neighbor is opposite class)

Error rate: $\boxed{\frac{4}{9}}$

(2) Compute the *leave-one-out cross-validation* error rate of a 3-Nearest-Neighbor classifier.

Due to alternation of labels, all 3-NN predictions are wrong:

Error rate: $\boxed{\frac{9}{9} = 1}$

(3) Compute the *leave-one-out cross-validation* error rate of an 8-Nearest-Neighbor classifier.

This classifier predicts $y = -1$ in all cases;

Error rate: $\boxed{\frac{4}{9}}$

This page is intentionally blank, use as you wish.

Name: THE SOLUTION ID#:

Problem 4 Regression via Gradient Descent *(11 points)*

Suppose that you would like to predict a positive number y from a one-dimensional input feature x. You think that the relationship between y and x is well approximated by an exponential function, $\hat{y}(x) = \exp(ax + b)$. Given m training examples $(x^{(i)}, y^{(i)})$, you want to find regression parameters a, b that minimize the *mean squared error* (MSE) in your predictions:

$$J(a,b) = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}(x^{(i)})\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \exp(ax^{(i)} + b)\right)^2.$$

When computing gradients below, remember that $\frac{\partial}{\partial u}u^2 = 2u$ and $\frac{\partial}{\partial u}\exp(u) = \exp(u)$.

(1) What is $\frac{\partial}{\partial a}J(a,b)$, the derivative of the MSE with respect to parameter a?

$$\frac{\partial}{\partial a}J(a,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial a}\left(y^{(i)} - \exp(ax^{(i)}+b)\right)^2$$

$$= \frac{-2}{m}\sum_{i=1}^{m}\left(y^{(i)} - \exp(ax^{(i)}+b)\right)\exp(ax^{(i)}+b)x^{(i)}$$

(2) What is $\frac{\partial}{\partial b}J(a,b)$, the derivative of the MSE with respect to parameter b?

$$\frac{\partial}{\partial b}J(a,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial b}\left(y^{(i)} - \exp(ax^{(i)}+b)\right)^2$$

$$= \frac{-2}{m}\sum_{i=1}^{m}\left(y^{(i)} - \exp(ax^{(i)}+b)\right)\exp(ax^{(i)}+b)$$

(3) Suppose that at some iteration of gradient descent, the regression parameters have values $a = 2$, $b = 5$. Evaluating the derivatives of the MSE at this point, you find that $\frac{\partial}{\partial a}J(a, b) = -2$ and $\frac{\partial}{\partial b}J(a, b) = +3$. After one iteration of gradient descent with step size (learning rate) 0.5, what will be the new values of the a and b parameters?

To minimize, take step in <u>negative</u> gradient:

$$a^{new} = a - 0.5\left(\frac{\partial J(a,b)}{\partial a}\right) = 2 - 0.5(-2) = \boxed{3}$$

$$b^{new} = b - 0.5\left(\frac{\partial J(a,b)}{\partial b}\right) = 5 - 0.5(+3) = \boxed{3.5}$$

(4) Is the single iteration of gradient descent computed in the previous part guaranteed to reduce the training MSE? Briefly justify your answer.

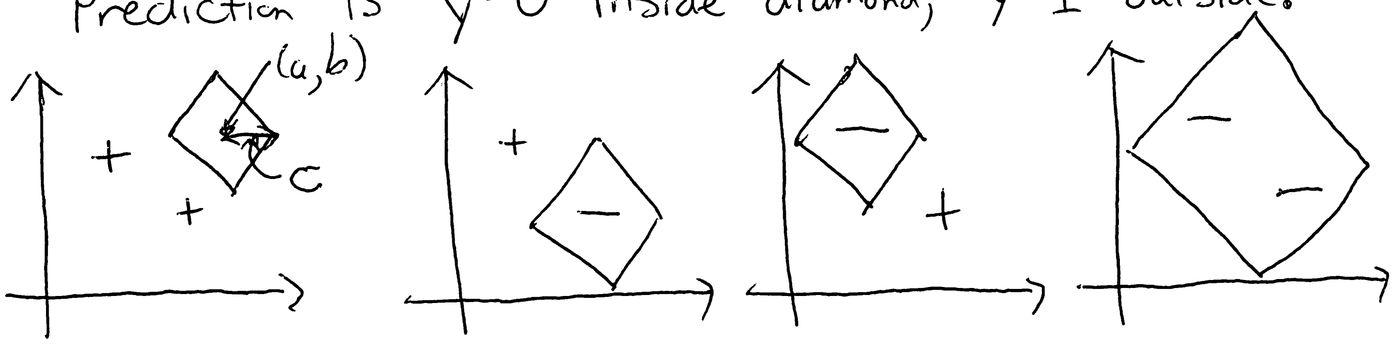
$\boxed{\text{No.}}$ Guaranteed to reduce error for sufficiently small step size, but 0.5 could be too large.

Problem 5 VC Dimension *(10 points)*

Consider the problem of learning a binary classifier that depends on two real-valued features x_1, x_2. The classifier takes the form $\hat{y}(x) = T(|x_1 - a| + |x_2 - b| + c)$, where $T(z)$ is the threshold function: $T(z) = 1$ for $z \geq 0$, and $T(z) = 0$ for $z < 0$. The classifier parameters a, b, c are real-valued scalars.

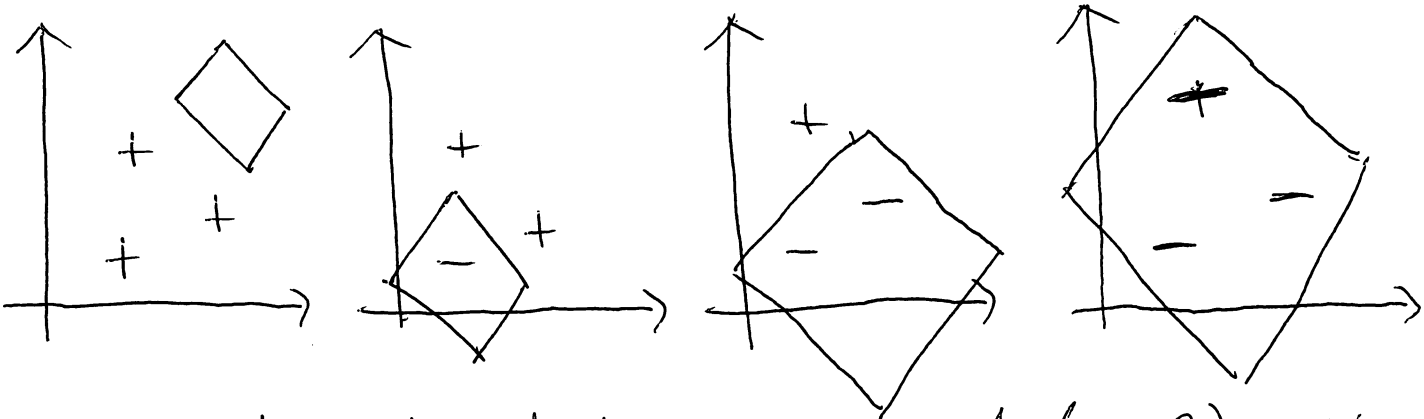
(1) Using four pictures, show that classifier $\hat{y}(x)$ can shatter all binary labelings of **2 data points**.

Decision boundary is a diamond centered at (a,b).
(Similar to L_1 regularizer from lecture.)
Prediction is $\hat{y}=0$ inside diamond, $\hat{y}=1$ outside.



(2) Does there exist a set of **3 data points** that classifier $\hat{y}(x)$ can shatter? If yes, give an example and briefly explain why. If no, give an example labeling that cannot be separated.

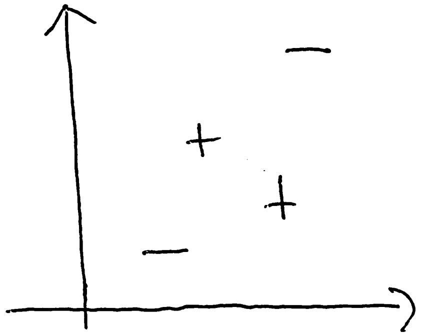
YES. Examples of how to shatter all labelings:



Can draw diamond that contains any 1 (or 2) points, and excludes any 2 (or 1) points.

(3) Does there exist a set of **4 data points** that classifier $\hat{y}(x)$ can shatter? If yes, give an example and briefly explain why. If no, give an example labeling that cannot be separated.

No. Consider this set of points:



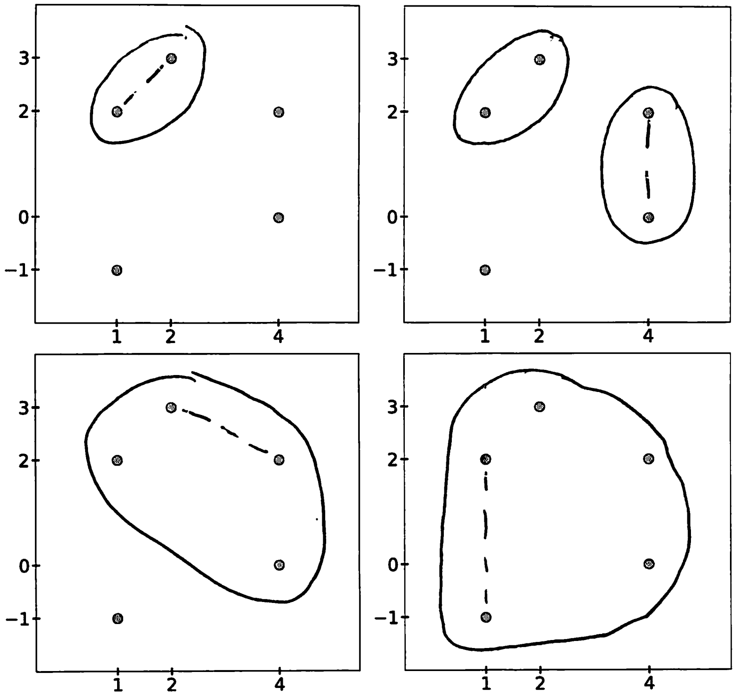
There is no diamond that contains both negative points, and excludes both positive points.

(4) Based on your results in the previous parts, what do you think the VC dimension of $\hat{y}(x)$ equals? Briefly justify your answer. (A formal proof is not required.)

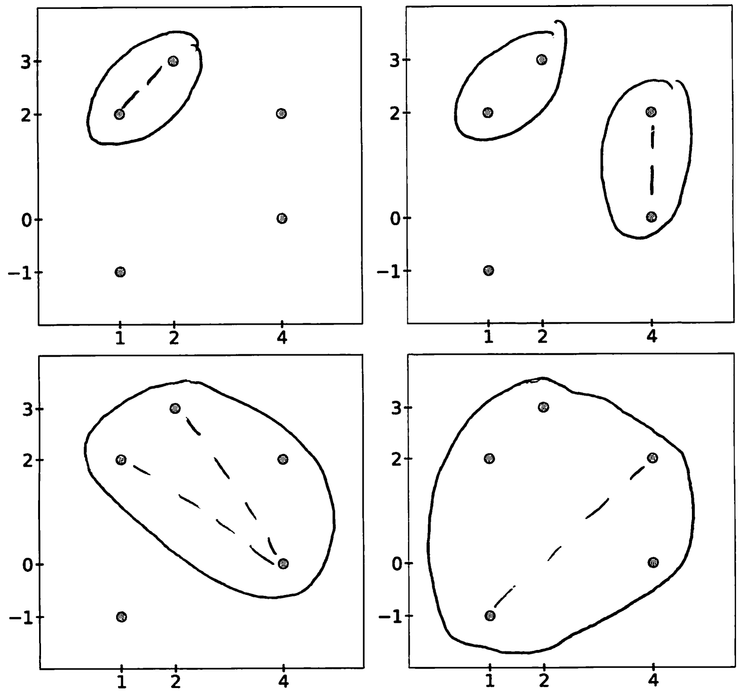
3 Can shatter 3 data points, but not 4. (This equals the number of parameters, but that is not sufficient justification.)

Problem 6 Clustering Algorithms *(12 points)*

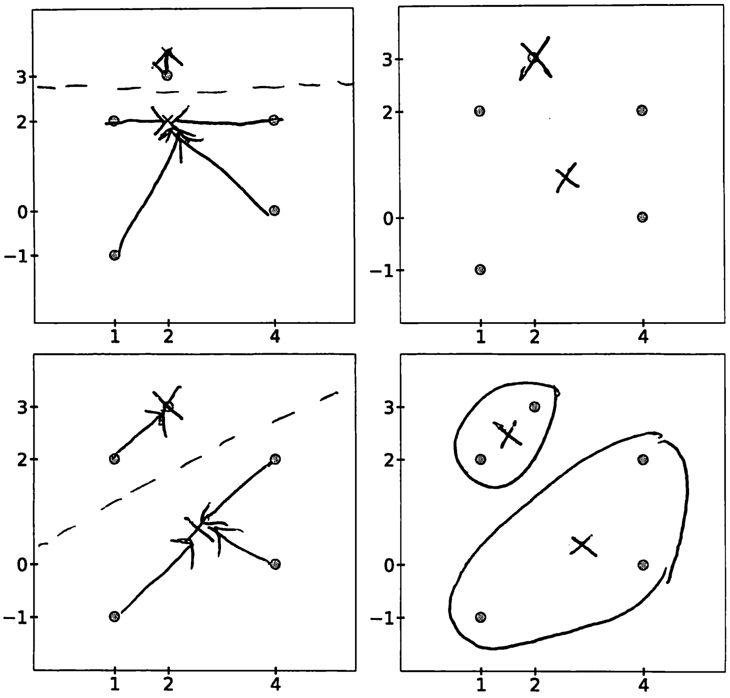
(1) Execute the hierarchical agglomerative clustering algorithm on the 5 data points below, using the **complete linkage** (maximum distance) criterion for cluster merges. Stop when the algorithm would terminate, or after 4 iterations, whichever comes first. Show the clusters after each iteration in a separate figure below.



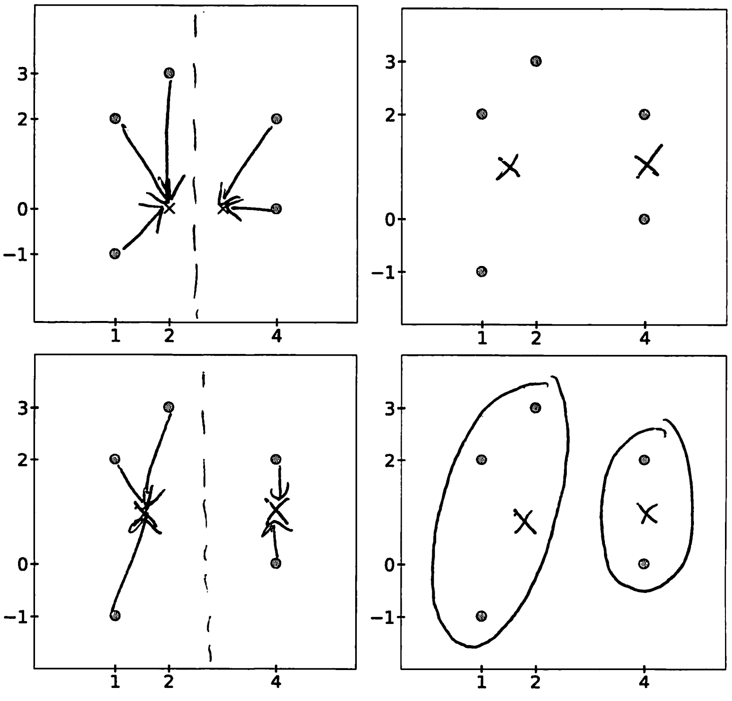
(2) Execute the hierarchical agglomerative clustering algorithm on the 5 data points below, using the **single linkage** (minimum distance) criterion for cluster merges. Stop when the algorithm would terminate, or after 4 iterations, whichever comes first. Show the clusters after each iteration in a separate figure below.



(3) Execute the K-means clustering algorithm on the 5 data points below, initializing with the $K = 2$ cluster means plotted as crosses. Stop when the algorithm would terminate, or after 4 iterations, whichever comes first. Show the cluster means and cluster assignments after each step of the algorithm in a separate figure below.



(4) Execute the K-means clustering algorithm on the 5 data points below, initializing with the $K = 2$ cluster means plotted as crosses. Stop when the algorithm would terminate, or after 4 iterations, whichever comes first. Show the cluster means and cluster assignments after each step of the algorithm in a separate figure below.



Name: THE SOLUTION ID#:

Problem 7 Multiple Choice *(10 points)*

(1) **TRUE** or [✓ **FALSE**] When learning a decision tree, reducing the *minParent* parameter (the minimum amount of data required to split a node) is a way to reduce overfitting.

Reducing minParent increases complexity.

(2) [✓ **TRUE**] or **FALSE** : Using bagging to build an ensemble of several classifiers is a way to reduce overfitting.

(3) **TRUE** or [✓ **FALSE**] When training a model using boosting, the different classifiers in the ensemble may be learned in parallel to improve computational efficiency.

Boosting sequentially trains to correct errors.

(4) [✓ **TRUE**] or **FALSE** : For linear regression models, L_1 (lasso) regularization is a way to select the most important subset of the input features.

(5) **TRUE** or [✓ **FALSE**] The computational complexity of agglomerative clustering algorithms scales linearly with the amount of training data.

Quadratic

(6) [✓ **TRUE**] or **FALSE** : The expectation maximization (EM) algorithm for Gaussian mixture models can be used to learn clusters whose shape is not circular.

(7) **TRUE** or [✓ **FALSE**]: The expectation maximization (EM) algorithm for learning Gaussian mixture models always converges to the global maximum of the data log-likelihood.

Local maximum.

(8) [✓ **TRUE**] or **FALSE** : The global minimum of the principal component analysis (PCA) objective may be computed via a singular value decomposition (SVD) of the training data.

(9) **TRUE** or [✓ **FALSE**]: Reinforcement learning algorithms are given a control policy, and learn an optimal reward function from training data.

Given rewards, learn policy.

(10) [✓ **TRUE**] or **FALSE** : Given a fixed policy for a Markov decision process, the value of each state can be computed by solving a linear system of equations.

This page is intentionally blank, use as you wish.