FINAL

CS273a ~~Midterm~~ Exam

Introduction to Machine Learning: Fall 2012

Tuesday December 11th. 2012

Your name: SOLUTIONS

Name of the person in front of you (if any):

Name of the person to your right (if any):

- Total time is 1:~~15~~ 50. 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 to come over.
- Turn in any scratch paper with your exam.

Problem 1: VC Dimension (10p)

Argue by example / counterexample what is the VC dimension of each of the following classifiers.

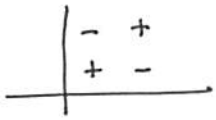
(5p) (a) A perceptron on two *binary* features

Binary features $\Rightarrow x_i \in \{0, 1\}$

VCDim = 3

Can shatter 3 points

Cannot shatter 4 points

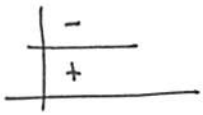


(5p) (b) A decision stump on two *binary* features

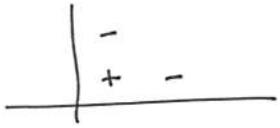
Binary features $\Rightarrow x_i \in \{0, 1\}$.

VC Dim = 2

Can shatter 2 points:



cannot shatter 3



(No axis-aligned split can separate)

Problem 2: Decision Trees (12p)

We plan to use a decision tree to predict an outcome y using four features, $x_1, \ldots, x_3$. We observe six training patterns, each of which we represent as $[x_1, x_2, x_3]$ (so, "010" means $x_1 = 0$, $x_2 = 1$, $x_3 = 0$). We observe the training data.

$y = 0$: [100], [111], [001]

$y = 1$: [110], [110], [011]

You may find the following values useful (although you may also leave logs unexpanded):

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

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

(3p) (a) What is the entropy of y?

$3/6 \log 6/3 + 3/6 \log 6/3 = 1$ bit

(3p) (b) Which variable would you split first? Justify your answer.

	X_1		X_2		X_3	
	=0	=1	=0	=1	=0	=1
y=0	001	100 111	100 001	111	100	111 001
y=1	011	110 110	—	110 110 011	110 110	011

By inspection, X_2 has the lowest expected entropy ($\Rightarrow$ higher info gain)

$H(y|x_1) \geq H(y|x_3) \geq H(y|x_2)$

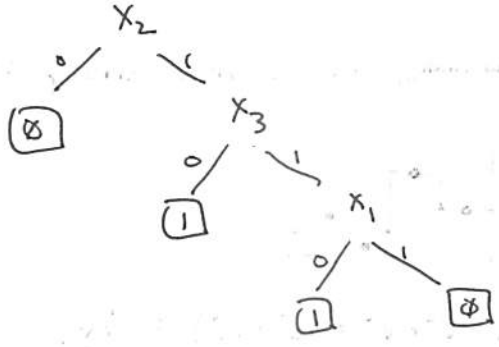
(3p) (c) What is the information gain of the variable you selected in part (b)?

$H(y) - [2/6\ H(\emptyset) + 4/6\ H(1/4)]$

$= 1 - [\emptyset + 2/3\ (1/4 \log 4 + 3/4 \log 4/3]$

$= 1 - 1/6 \log 4 - 1/2 \log 4/3 = 2/3 - 1/2 \lg(.41)$

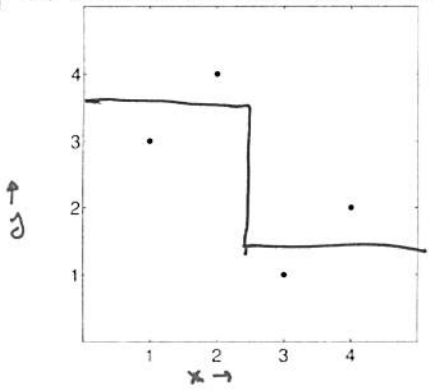
(3p) (d) Draw the rest of the decision tree learned on these data.



Problem 3: Gradient Boosting (9p)

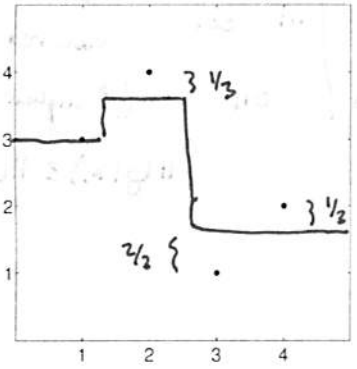
Consider the following data set consisting of four points; for convenience, the data are repeated in each part.

(3p) (a) Compute the best single decision stump regressor function, to minimize mean squared error.



(this predictor has MSE = $\frac{1}{4} \cdot (4 \cdot \frac{1}{2}^2) = \frac{1}{4}$.)

(3p) (b) Now, we wish to create a gradient boosted ensemble of decision stumps to minimize MSE. Starting from the decision stump learned in 9a), and using a learning rate of 1, what is the next predictor? Show your work.



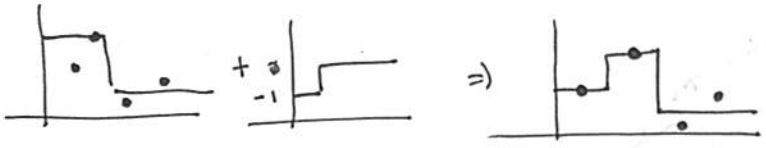
After (a), we have error residual

If we fit a decision stump, we get  ⇒ sum at left.

$MSE = \frac{1}{4} \cdot (0^2 + \frac{1}{3}^2 + \frac{2}{3}^2 + \frac{1}{3}^2) = \frac{3}{18}$.

(3p) (c) Is the resulting ensemble the best possible ensemble of two decision stumps? If yes, why? If not, give a better ensemble.

No - if you do not train them sequentially, you can get a better predictor, eg



⇒ MSE is $\frac{1}{4}\left(0^2 + 0^2 + \frac{1}{2}^2 + \frac{1}{2}^2\right) = \frac{1}{8}$.

Problem 4: Naïve Bayes (10p)

We plan to use a naïve Bayes classifier to predict an outcome y using four features, $x_1, \ldots, x_3$. We observe five training patterns, each of which we represent as $[x_1, x_2, x_3]$ (so, "010" means $x_1 = 0$, $x_2 = 1$, $x_3 = 0$). We observe the training data,

$y = 0$: [000], [111]
$y = 1$: [100], [010], [001]

(4p) (a) Compute (& show) all the necessary probabilities for a naïve Bayes model.

$p(y) = 3/5$

$p(x_1 | y=0) = 1/2$ $\quad p(x_1 | y=1) = 1/3$

$p(x_2 | y=0) = 1/2$ $\quad p(x_2 | y=1) = 1/3$

$p(x_3 | y=0) = 1/2$ $\quad p(x_3 | y=1) = 1/3$

(3p) (b) Suppose you observe $x = [110]$. What class (value of y) would you predict? Show your work.

$p(y=0)\, p(\underline{x} | y=0) = 2/5 \cdot 1/2 \cdot 1/2 \cdot 1/2 = 1/20$

$p(y=1)\, p(\underline{x} | y=1) = 3/5 \cdot 1/3 \cdot 1/3 \cdot 2/3 = 2/45$ $\Rightarrow$ predict $y = 0$.

(3p) (c) Suppose you observe $x = [100]$. Compute the posterior probability, $p(y|x = 100)$.

$p(y=0)\, p(\underline{x} | y=0) = 2/5 \cdot 1/2 \cdot 1/2 \cdot 1/2 = 1/20 = \frac{9}{5 \cdot 2 \cdot 2 \cdot 3 \cdot 3}$

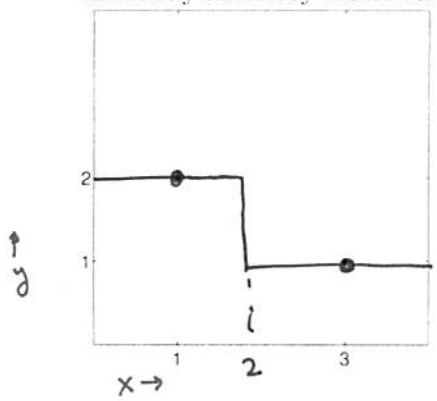
$p(y=1)\, p(\underline{x} | y=1) = 3/5 \cdot 1/3 \cdot 2/3 \cdot 2/3 = 4/45 = \frac{4 \cdot 4}{5 \cdot 3 \cdot 3 \cdot 2 \cdot 2}$

$\Rightarrow p(y=1 | \underline{x}) = \frac{16}{16+9} = \frac{16}{25}$.

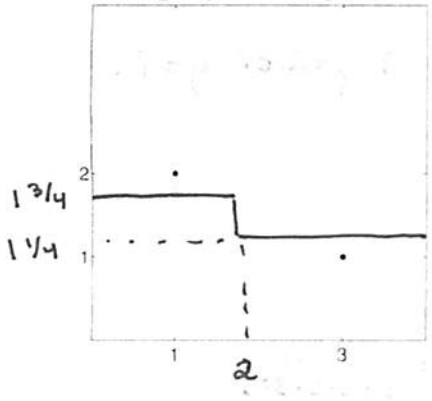
Problem 5: Bagging (9p)

Consider the data set, consisting of two data points, given in each part.

(3p) (a) Draw the regression function (predicted values for all x) using a nearest-neighbor regressor. Label any necessary values on your graph.



(4p) (b) Suppose that we create a very large ensemble of *bagged* nearest-neighbor regressors, using data set draws of size two. Compute the regression function of the complete ensemble, again labeling any necessary values.



There are four possible draws ⇒ 3 unique data sets:

{a,a}: occurs 1/4

{b,b}: 1/4

{a,b} = {b,a}: 1/2 of the time.

⇒ ensemble average predicts nearest point 3/4 of members; other point 1/4 of member ⇒ graph at left

(2p) (c) Is the model in (b) simpler or more complex than the model in (a)? Why?

Simpler: (1) Bagging tends to reduce complexity / overfitting

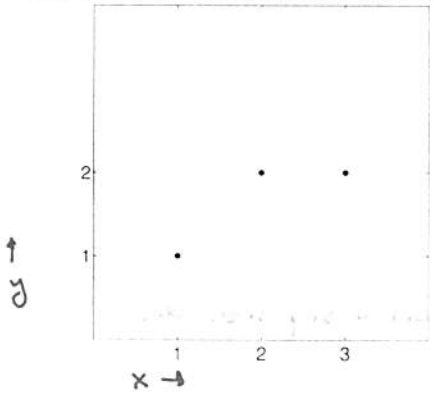
- you can see training error has gone up (& the data are no longer memorized)

(2) The function is "simpler" - it is closer to a constant predictor.

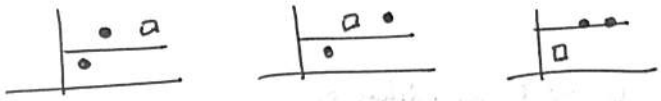
Problem 6: Cross-validation (8p)

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

(4p) (a) Compute the leave-one-out cross-validation error of a zero-order (constant) predictor.



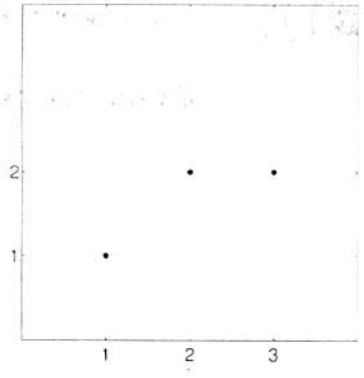
Leave out each data point ⇒



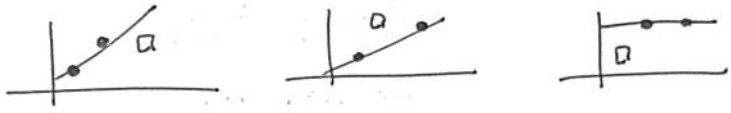
$(1/2)^2 + (1/2)^2 + (1)^2$

$\Rightarrow 1/3 \left[(1/2)^2 + (1/2)^2 + 1^2 \right) = 1/2.$

(4p) (b) Compute the leave-one-out cross-validation error of a first-order (linear) predictor.



Leave out each ⇒



$1^2 + (1/2)^2 + 1^2$

$\Rightarrow 1/3 \left[1^2 + 1/2^2 + 1^2 \right) = 3/4.$

Problem 7: Latent space models (8p)

Suppose that, as in HW5, we wish to model a collection of text documents using a latent space model. For interpretability, we would like our latent representation to be non-negative. As one solution, we use an exponential transform to ensure positive values, giving the model

$$x_j^{(i)} \approx \sum_k \exp(U_{ik}) \exp(V_{kj})$$

Give a stochastic gradient descent algorithm to learn this model, minimizing the mean squared error in the predicted values. Include all necessary details for the implementation.

A simple SGD algorithm is

1. Initialize U, V to something at random
2. Choose a stopping criterion (eg. # of steps) and a step size α.
3. while (!stop) {

```
for i = 1... m
  for j = 1..d
    E = (x_j^(i) - Σ_k exp(U_ik) exp(V_kj));   // compute signed error.
    Ũ = U;  Ṽ = V;                             // save old values
    for k = 1.. K.
      U_ik ← U_ik − α ∇_{U_ik} J
      V_ik ← V_ik − α ∇_{V_kj} J
    end
  end
end
```

where $J(\cdot) = \left(x_j^{(i)} - \sum_k \exp(u)\exp(v)\right)^2$ is the squared error loss

and

$$\nabla_{U_{ik}} J = -E \cdot \exp(\tilde{U}_{ik}) \exp(\tilde{V}_{kj})$$

$$\nabla_{V_{kj}} J = -E \cdot \exp(\tilde{U}_{ik}) \exp(\tilde{V}_{kj})$$

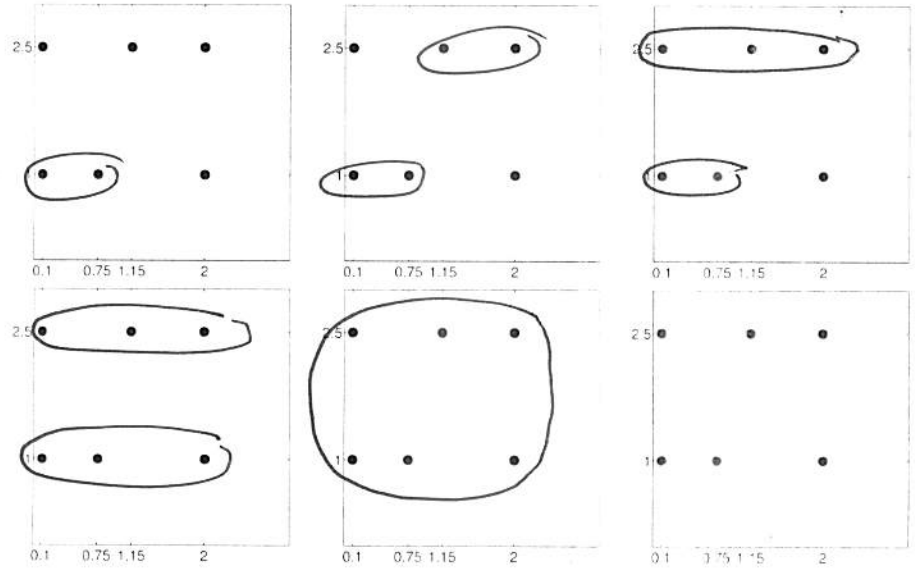
are the derivatives with respect to each element of u, v.

notation should really be $\frac{\partial}{\partial U_{ik}}$ and $\frac{\partial}{\partial V_{kj}}$ instead

Problem 8: Clustering (10p)

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

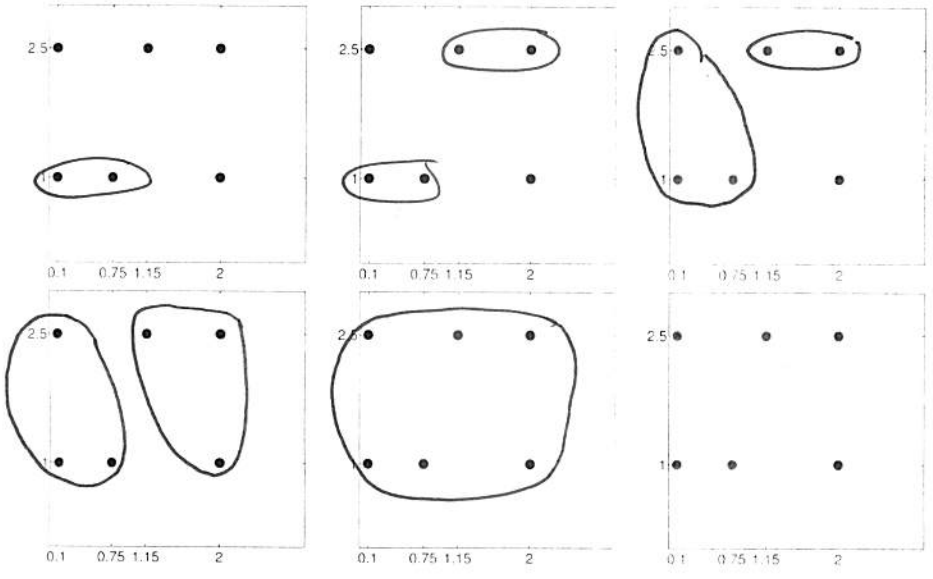
(5p) (a) Execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "single linkage" for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.



Join nearest clusters.

Cluster distance = nearest points in the two clusters.

(5p) (b) Now execute hierarchical agglomerative clustering on the data points, but use "complete linkage" for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.



Join nearest two clusters.

Cluster distance = furthest distance between two points in the clusters.

From board:

$\sqrt{(.85)^2 + (1.5)^2} = 1.724$

$\sqrt{(.65)^2 + (1.5)^2} = 1.635$

CS273a Final Exam

Introduction to Machine Learning: Fall 2013

Thursday December 12th, 2013

Your name: SOLUTIONS

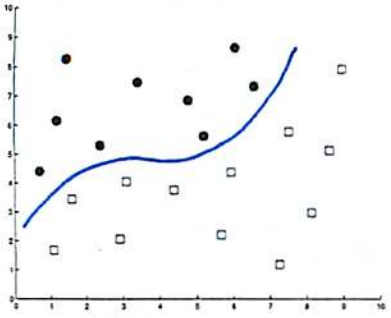
Your UCInetID (all caps):

Your Seat (row and number):

- Total time is 1:50. READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Closed book; one page of (your own) notes
- Please **write clearly** and **show all your work.**
- If you need clarification on a problem, please raise your hand and wait for the instructor to come over.
- Turn in any scratch paper with your exam.

Problem 1: Separability

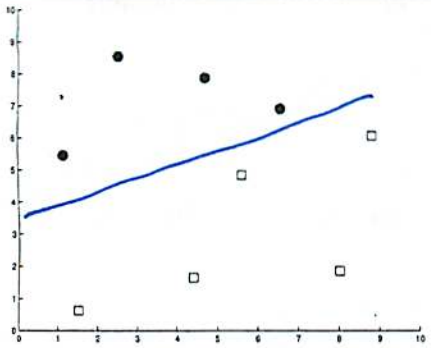
For each of the following examples of training data, sketch a classification boundary that separates the data. State whether or not the data are linearly separable, and if not, give a set of features that would allow the data to be separated.



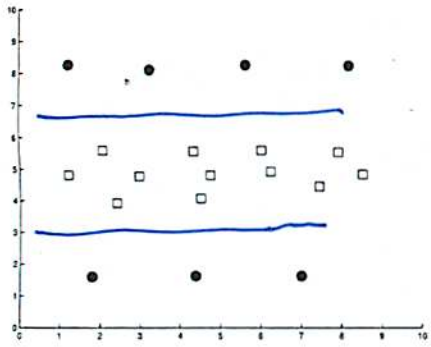
No, not linearly separable.

Based on the curve, $x_2 = a + bx_1 + cx_1^2 + dx_1^3$

the features $[1 \; x_1 \; x_1^2 \; x_1^3 \; x_2]$ should work.



Yes, linearly separable.



No, not linearly separable.

The decision boundary can be obtained as

$ax_2^2 + bx_2 + c = 0$

$\Rightarrow [1 \; x_2 \; x_2^2]$ is enough.

Problem 2:

Select the best choice to complete each statement.

Increasing the number of hidden nodes in a neural network will most likely
(increase **decrease** not change) the bias.

Decreasing regularization on the weights in logistic regression will most likely
(increase decrease **not change**) the VC dimension.

Increasing the amount of data will most likely
(increase **decrease** not change) the variance.

Increasing the regularization on a perceptron will most likely
(**increase** decrease not change) the bias.

Decreasing the maximum depth of a decision tree will most likely
(increase **decrease** not change) the VC dimension.

Increasing the depth of a decision tree will most likely
(increase **decrease** not change) the bias.

Possible argument for no change - if depth is already very large compared to the amt of data

The predictions of a k-nearest neighbor classifier
(**will** will not) be affected by pre-processing to normalize the data.

Reducing the number of features using PCA will most likely
(increase **decrease** not change) the variance.

Linear regression
(**can** cannot) be solved using either matrix algebra or gradient descent.

The predictions of a regression tree
(will **will not**) be affected by pre-processing to normalize the features.

Problem 3: Regression

Suppose that we train a *non-linear* regression model on m data, where our prediction is

$$\hat{y}(x) = a + \exp(bx)$$

for two scalar parameters a, b.

(a) Write down the formula for the mean-squared error on the training data, and compute its gradient with respect to the parameters.

$$MSE = J(a,b) = \frac{1}{m}\sum_i \left(y^i - \hat{y}(x^i)\right) = \frac{1}{m}\sum_i \left(y^i - a - \exp(bx^i)\right)^2$$

$$\nabla J = \left[\frac{\partial J}{\partial a} \quad \frac{\partial J}{\partial b}\right]$$

$$\frac{\partial J}{\partial a} = \frac{1}{m}\sum_i \left(y - \hat{y}(x^i)\right)(-1). \qquad (1)$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_i \left(y^i - \hat{y}(x^i)\right)\cdot(-1)\exp(bx^i)\cdot b. \qquad (2)$$

(b) Give pseudo-code for (batch) gradient descent on this problem. Be sure to specify initialization, the update itself (in enough detail to enable coding), and a stopping condition (again, in enough detail to enable coding).

```
Init a,b :  a=0.  b=0.   a'=inf  b'=inf.   α = stepsize.
while (~done)
    for i=1..m ,  ŷ(x^i) = a+exp(b x^i)
    a ← a - α ∂J/∂a        as in (1)
    b ← b - α ∂J/∂b        as in (2)        % take a step (could update stepsize)
    done = [ (a-a')^2 + (b-b')^2 < ε ]      % check for convergence
    a = a' ; b = b' ;                       % save old values
```

(c) Give at least one advantage of batch gradient descent over stochastic gradient descent. In contrast, when would using stochastic gradient be more appropriate?

Batch - always a descent on J (for sufficiently small α)

⇒ easy to debug, easy to assess convergence, monotonic.

SGD - more appropriate than batch when m is very large (many data)

- in this case, batch updates will be very slow. (all data processed before each step).

Problem 4: Naïve Bayes

Consider the following table of measured data:

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

We will use the three observed features x_1, x_2, x_3 to predict class y. In the case of a tie, we will prefer to predict class $y = 1$.

(a) Write down the probabilities necessary for a naïve Bayes (NB) classifier:

$\Pr[y=1] = 3/7$

$\Pr[x_1=1 \mid y=0] = 3/4$ $\quad \Pr[x_2=1 \mid y=0] = 3/4$ $\quad \Pr[x_3=1 \mid y=0] = 1/2$

$\Pr[x_1=1 \mid y=1] = 1/3$ $\quad \Pr[x_2=1 \mid y=1] = 2/3$ $\quad \Pr[x_3=1 \mid y=1] = 1.$

(b) Using your NB model, what value of y is predicted given observation $(x_1, x_2, x_3) = (000)$.

Predict $y = \emptyset$. $\quad (y=1$ has $\Pr[x_3=0 \mid y=1] = \emptyset)$.

(c) Using your NB model, what is the probability $p(y = 1|x_1 = 1, x_2 = 1, x_3 = 1)$?

$$= \frac{3/7 \cdot 1/3 \cdot 2/3 \cdot 1}{(\text{"}) + 4/7 \cdot 3/4 \cdot 3/4 \cdot 1/2} = \frac{6/9}{6/9 + 9/8} = \frac{48}{48+81} = \frac{48}{129}$$

(d) Using your NB model, what is the probability $p(y = 1|x_1 = 0)$?

$$= \frac{3/7 \cdot 2/3}{(\text{"}) + 4/7 \cdot 1/4} = \frac{2/7}{2/7 + 1/7} = 2/3.$$

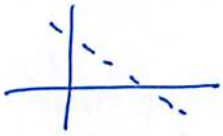
Problem 5: Perceptrons and VC Dimension

In this problem, consider the following perceptron model on two features:

$$\hat{y}(x) = \text{sign}(\, w_0 + w_1 x_1 + w_2 x_2 \,)$$

and answer the following questions about the decision boundary and the VC dimension.

(a) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier



Any hyperplane - eg, decision boundary is an arbitrary line.

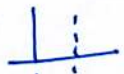
(b) What is its VC dimension?

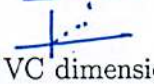
3 - VC dim of a perceptron is $d+1$, and $d=2$.

Now suppose that I also enforce an additional condition on the parameters of the model: that **at most two** of the weights w_i are non-zero (so, at least one weight is zero).

(c) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier

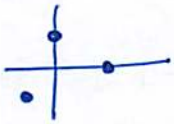
If $w_1 = 0$,  horizontal decision

$w_2 = 0$  vertical decision

$w_0 = 0$  line through the origin

} Decision boundary can be an arbitrary horizontal or vertical split, or a line through the origin.

(d) What is its VC dimension?

Still 3:  can still be shattered.

(2, can't be higher than part (b))

Finally, I enforce that **at most one** of the weights w_i is non-zero (so, at least two are zero).

(e) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier

$w_0 \neq 0$ - entire plane same decision

$w_1 \neq 0$ - right vs left half-plane

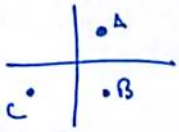
$w_2 \neq 0$ - upper vs lower half plane.

} union of these functions.

(f) What is its VC dimension?

Now 2 - can't shatter above example; not possible to arrange points to get all three ++- patterns...

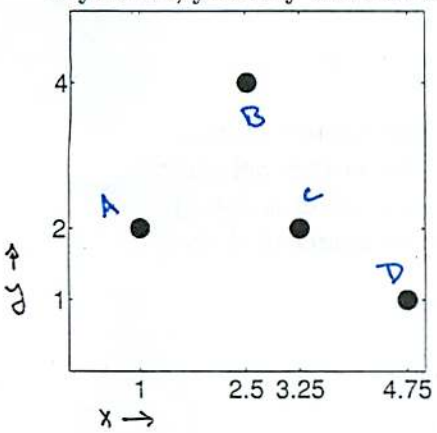
↳ (can't put points on the axes, though).



⇒ can't predict A = C = +1, B = -1.

Problem 6: Cross-validation

Consider a regression problem for predicting the following data points, using the k-nearest neighbor regression algorithm from class and the homework to minimize mean squared error (MSE). (Note: if you like, you may leave an arithmetic expression, e.g., leave values as "$(.6)^2$".)

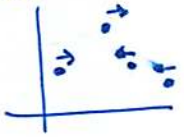


(a) For $k = 1$, compute the training error on the provided data.

ϕ.

(b) For $k = 1$, compute the leave-one-out cross-validation error on the data.

nearest nbr:



$\Rightarrow \frac{1}{4}\left[2^2 + 2^2 + 2^2 + 1^2\right] = 13/4$

(c) For $k = 3$, compute the training error on the provided data.

neighbors:	preds:
A: ABC	8/3
B: ABC	8/3
C: BCD	7/3
D: BCD	7/3

$\Rightarrow \frac{1}{4}\left[(2/3)^2 + (4/3)^2 + (1/3)^2 + (4/3)^2\right] = 37/36$

(d) For $k = 3$, compute the leave-one-out cross-validation error on the data.

A: BCD	7/3
B: ACD	5/3
C: ABD	7/3
D: ABC	8/3

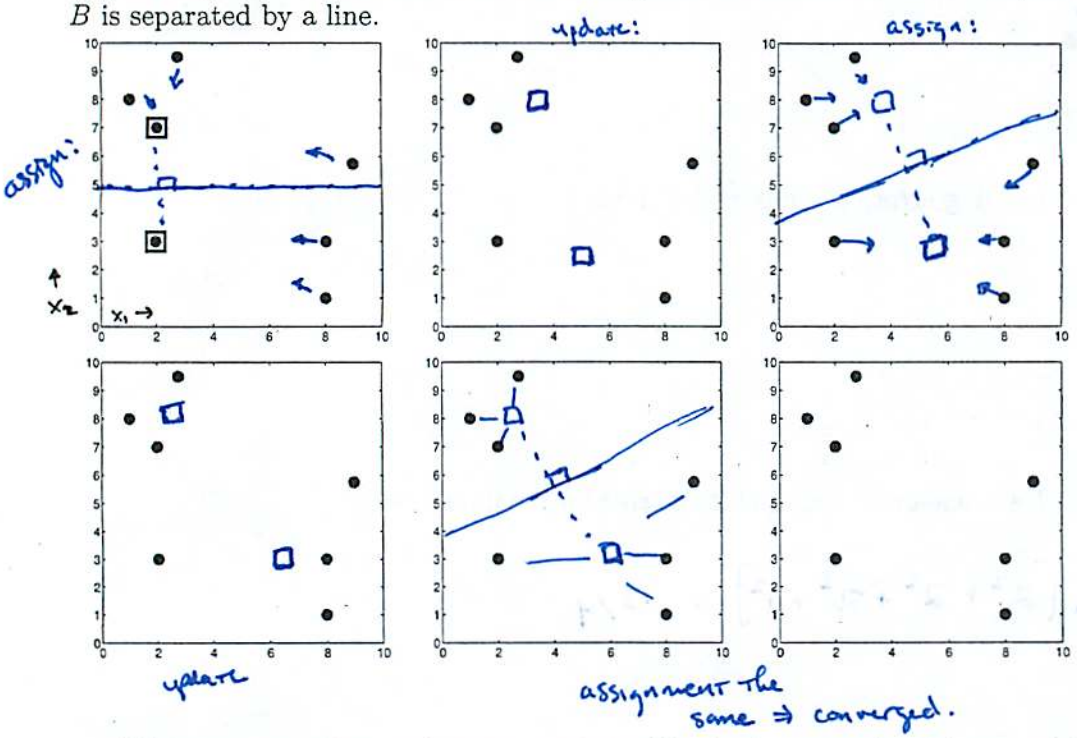
$\Rightarrow \frac{1}{4}\left[(1/3)^2 + (7/3)^2 + (1/3)^2 + (5/3)^2\right] = 76/36$

Problem 7: Clustering

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

k-means

(a) Starting from the two cluster centers indicated by squares, perform k-means clustering on the data points. In each panel, indicate (somehow) the data assignment, and in the next panel show the new cluster centers. Stop when converged, or after 6 steps (3 iterations), whichever is first. It may be helpful to recall from our nearest-neighbor classifier that the set of points nearer to A than B is separated by a line.



(b) Write down the cost function optimized by the k-means algorithm, explaining your notation.

Let μ_c be the center of cluster c.

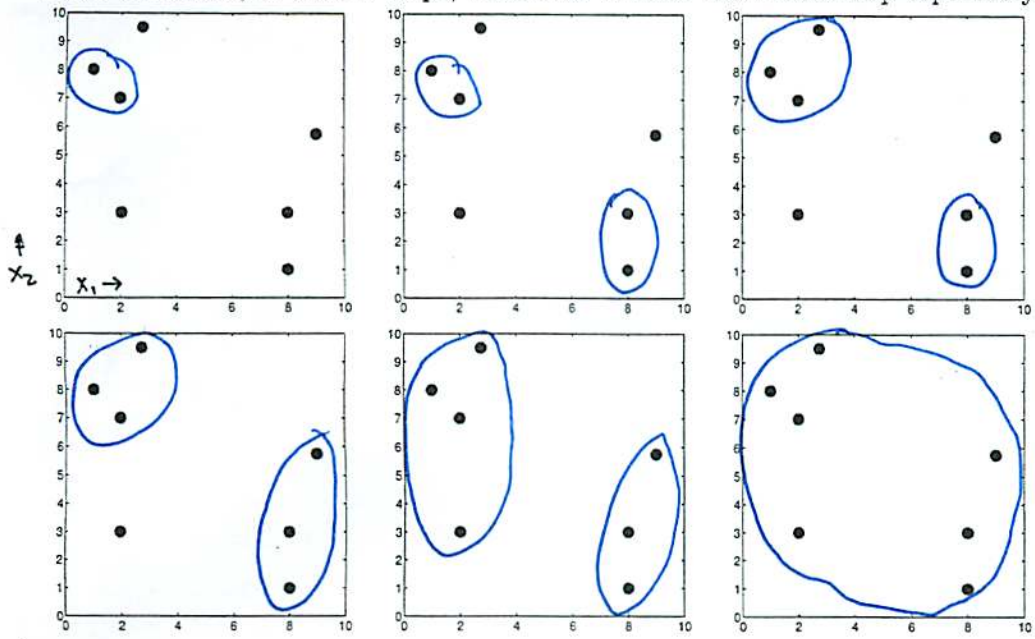
z_i be the cluster assignment of data point i.

$$J = \sum_{i=1}^{m} \|x^{(i)} - \mu_{z_i}\|^2$$

where $\|\cdot\|^2$ is the Euclidean distance / length squared.

Linkage

(a) Now execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "complete linkage" (maximum distance) for the cluster scores. Stop when the algorithm would terminate, or after 6 steps, whichever is first. Show each step separately in a panel.



Complete linkage ⇒ join nearest pair of clusters, where distance is given by farthest points.

(b) What is the algorithmic (computational) complexity of the hierarchical clustering algorithm? Briefly justify your answer.

$O(n^2)$.

Initial step calculates $\binom{n}{2} = O(n^2)$ pairwise distances

Each of n iterations (one join per step)

requires updating new cluster's distance to remaining clusters

⇒ $(n-2) + (n-3) + (n-4) + \cdots (1) = O(n^2)$

and, $O(n^2) + O(n^2) = O(n^2)$.

CS273 Final Exam

Introduction to Machine Learning: Winter 2015

Tuesday March 17th, 2015

Your name:

SOLUTIONS

Your UCINetID (e.g., myname@uci.edu):

Your seat (row and number):

- 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 to come over.
- Turn in any scratch paper with your exam.

(This page intentionally left blank)

Problem 1: (6 points) Multiple Choice

For the following questions, assume that we have m data points $y^{(i)}$, $x^{(i)}$, $i = 1 \ldots m$, each with n features, $x^{(i)} = [x_1^{(i)} \ \ldots \ x_n^{(i)}]$.

Circle one answer for each:

True or **false**: Early stopping can be used to reduce overfitting in neural networks.

True or **false**: The SVM learning algorithm will find the *globally optimal* model with respect to its objective function.

True or **false**: Increasing k in a k-nearest-neighbor classifier will decrease the bias.

True or **false**: Increasing the depth of a decision tree classifier will decrease the bias.

True or **false**: The VC dimension of a perceptron classifier is smaller than the VC dimension of a linear SVM.

True or **false**: If there exists a set of h instances that cannot be shattered by $f(x)$, then the VC dimension of f is less than h.

Problem 2: (4 points) Short Answer

Give one advantage of the dual (kernel) form of support vector machines over the primal (linear) form, **and** one advantage of the primal form over the dual.

Dual: may be easier to specify kernel similarity than features for linear classification
many kernels work in high / infinite dimensional feature spaces
better if # of features is very high (maybe)
∞ - d.m. kernel $\Rightarrow$ nonparametric predictor, often gets better as $m \to \infty$ (# data)

Primal: more efficient if $m \gg n$ (lots of data), in computation & model storage
Comp. & model storage are fixed (don't grow with m)
This also means it is often more efficient at test time (if $m \gg n$)
Easy to apply standard algorithms like SGD.

Problem 3: (12 points) Decision Trees

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
1	0	0	1
1	1	1	1
0	0	1	1
1	1	0	0
1	1	0	0
0	1	1	0

(a) What is the entropy of y?

1 bit

(b) Which variable would you split first? Justify your answer.

X_1: 0 → $y=0$: 011, $y=1$: 001; 1 → $y=0$: 110 110, $y=1$: 100 111

X_2: 0 → $y=0$: X, $y=1$: 100 001; 1 → $y=0$: 110 110 011, $y=1$: 111

X_3: 0 → $y=0$: 110 110, $y=1$: 101; 1 → $y=0$: 011, $y=1$: 111 001

(c) What is the information gain of the variable you selected in part (b)?

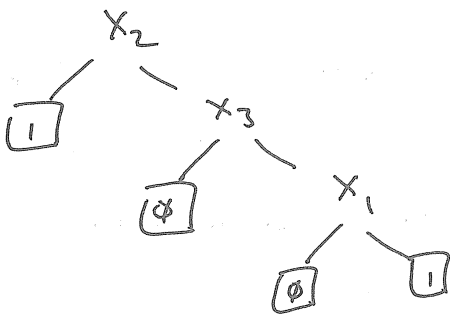
⇒ X_2.

$$IG = 1 - \left[\tfrac{1}{3} \cdot 0 + \tfrac{2}{3} \cdot \left(\tfrac{3}{4} \log \tfrac{4}{3} + \tfrac{1}{4} \log 4\right)\right]$$

$$= 1 - \tfrac{1}{2} \log \tfrac{4}{3} - \tfrac{1}{6} \log 4$$

$$= \tfrac{1}{2} \log 3 - \tfrac{1}{6} \log 4 \simeq .4591 \text{ bits}$$

(d) Draw the rest of the decision tree learned on these data.



X_3 next - gets 2 right (vs 1)

(can also calculate IG if desired, but clear by inspection)

Problem 4: (9 points) Gradient Descent & Latent Space Models

Suppose that, as in lecture, we wish to model a collection of text documents using a latent space model such as Latent Semantic Indexing. However, for interpretability, we would like our latent representation to be non-negative, so that each "direction" can only make a set of words more likely to appear, not less likely. As one solution, we use an exponential transform to ensure positive values, giving the model

$$x_j^{(i)} \approx \sum_k \exp(u_{ik}) \exp(v_{jk})$$

(so, data point i is a nonnegative linear combination u_i of nonnegative directions v_j). We wish to train our model (U, V) to minimize squared error from the obeserved data, and will train it using gradient descent.

(a) Write down an expression for J_{ij}, the mean squared error (MSE) of our model for element $x_j^{(i)}$, and for J, the overall MSE of our model.

$$J_{ij} = \left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right)^2$$

$$J = \frac{1}{m} \sum_{ij} J_{ij} \qquad \text{(also OK to normalize by } \frac{1}{mn}\text{)}$$

(b) Compute the derivatives of J_{ij} with respect to u_{ik} and v_{jk}.

$$\frac{\partial J_{ij}}{\partial u_{ik}} = 2\left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right) \cdot \left(-\exp(u_{ik}) \exp(v_{jk})\right)$$

(same!)

$$\frac{\partial J_{ij}}{\partial v_{jk}} = 2\left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right) \cdot \left(-\exp(u_{ik}) \exp(v_{jk})\right)$$

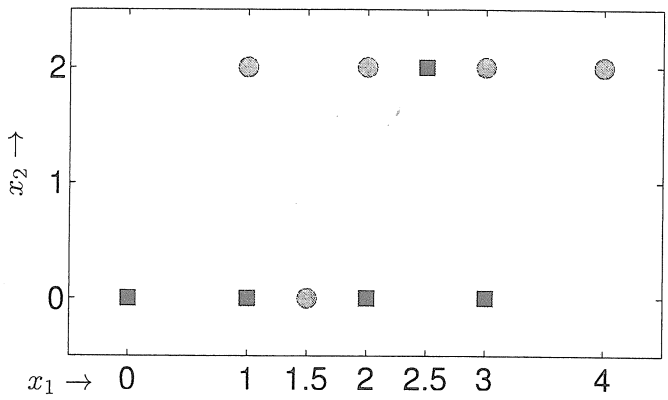
(c) Briefly describe how we could apply (e.g., give pseudocode for) stochastic gradient descent to learn u and v.

Init u, v to something (random usually)
Set stopping condition (eg # iterations) & step size α.
while (! done)
  for each i in random order
    for each j in random order (or do all j, depending on interpretation)
      $\hat{u} = u_i$, $\hat{v} = v$
      for $k = 1..K$
        $u_{ik} = u_{ik} - \alpha \frac{\partial J}{\partial u_{ik}}$
        $v_{jk} = v_{jk} - \alpha \frac{\partial J}{\partial v_{jk}}$ (use $\hat{u}, \hat{v}$ in ∂J to avoid overwriting)

Problem 5: (8 points) Cross-validation

Suppose that we learn a classifier on the following binary classification data. There are two real-valued features, x_1 and x_2, and a binary class $y \in \{0, 1\}$.

x_1	x_2	y
0	0	0
1	0	0
1.5	0	1
2	0	0
3	0	0
1	2	1
2	2	1
2.5	2	0
3	2	1
4	2	1



We decide to learn a decision tree as described in class. As in class, when the decision tree splits on the real-valued features, it puts the split threshold halfway between the data points on either side of the highest-scoring split. For example, if we first split on x_2, the algorithm would choose to split at $x_2 = 1$, which is halfway between the data at $x_2 = 0$ and $x_2 = 2$. In the case of ties, we prefer to predict class 0.

(a) What is the training error rate of a decision *stump* (decision tree with max depth 1, or two leaf nodes) trained on these data?

split on $x_2 = 1$

$\Rightarrow$ $2/10 = 1/5$ error

(b) What is the training error rate of a full decision tree (no maximum depth or pruning) trained on these data?

split until all correct

$\Rightarrow$ $\emptyset$ error

(c) What is the leave-one-out cross-validation error rate of a decision *stump* (decision tree with max depth 1) trained on these data?

All XV's split on $x_2 = 1$

$\Rightarrow$ $2/10 = 1/5$ error

(d) What is the leave-one-out cross-validation error rate of a full decision tree (no maximum depth) trained on these data?

All XVs split on $x_2 = 1$, Then at midpoint:

$\Rightarrow$ wrong at $x_2 = 0$, $x_1 = 1$, 1.5, 2

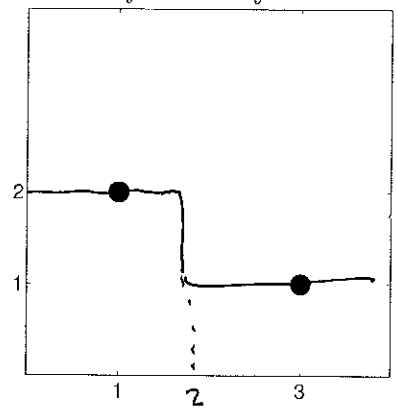
$x_2 = 2$, $x_1 = 2$, 2.5, 3

$\Rightarrow$ $6/10 = 3/5$ error.

Problem 6: (8 points) Bagging

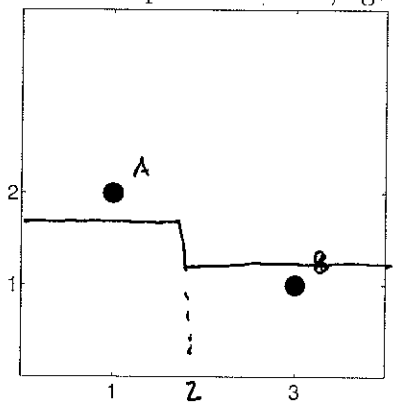
Consider a data set, consisting of two data points, plotted in each part.

(a) Draw the regression function (predicted values for all x) using a nearest-neighbor regressor. Label any necessary values on your graph.



$f(x) = \begin{cases} 2 & x < 2 \\ 1 & \text{ow.} \end{cases}$

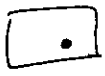
(b) Suppose that we create a very large ensemble of *bagged* nearest-neighbor regressors, using data set draws of size $m = 2$ during the boostrap sampling. Compute the regression function of the complete ensemble, again labeling any necessary values.



Bootstrap sampling draws data sets with replacement, which means we will get one of three possible sets, with probabilities:

$D = \{A, A\}$ — 1/4 of bootstraps ⇒ predict 2

$D = \{A, B\}$ — 1/2 of bootstraps ⇒ predict as in (a).



$D = \{B, B\}$ — 1/4 of bootstraps ⇒ predict 1

The bagged ensemble averages the predictions of its members ⇒ $1/4 \cdot 2 + 1/4 \cdot 1 + 1/2 \cdot \begin{cases} 2 & x < 2 \\ 1 & \text{ow.} \end{cases}$

$= \begin{cases} 1\,3/4 & x < 2 \\ 1\,1/4 & \text{ow.} \end{cases}$

(c) How does the training error in model (b) compare to the training error in (a)? (Note: you don't need to have answered (b) to answer this part.)

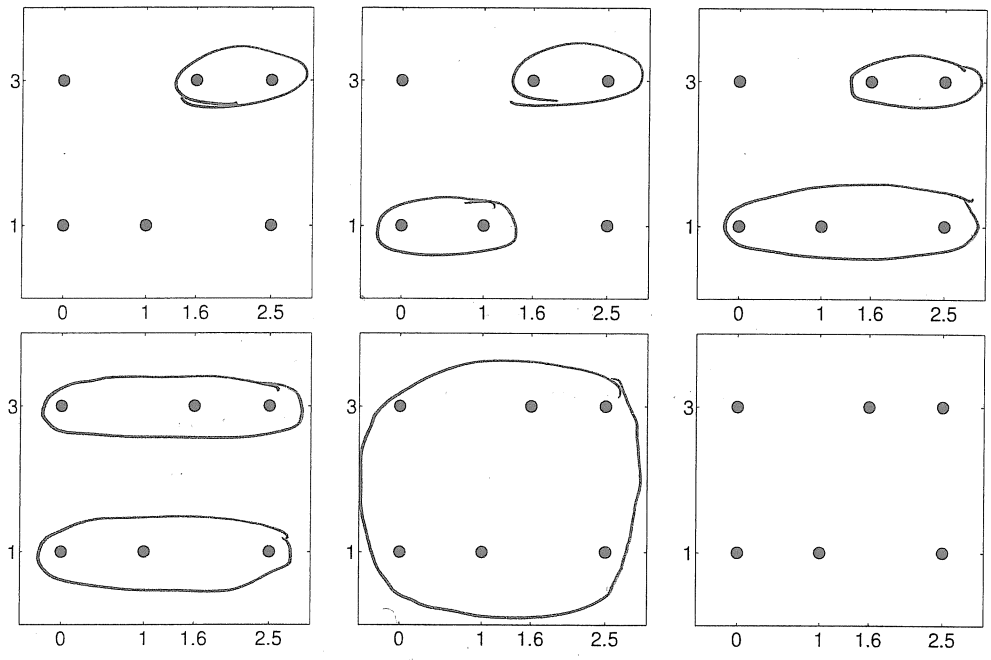
Training error has increased (MSE was 0, now $(1/4)^2$)

- our bagged model is overfitting less than the original; it is unable to "memorize" the data points.

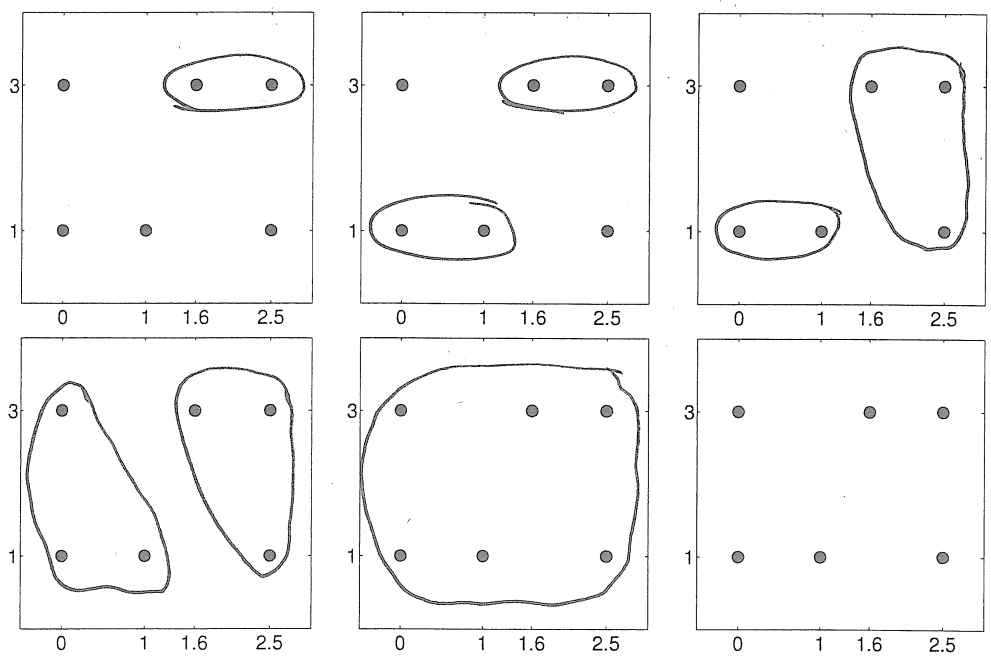
Problem 7: (8 points) Clustering

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

(a) Execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "single linkage" (minimum distance) for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.

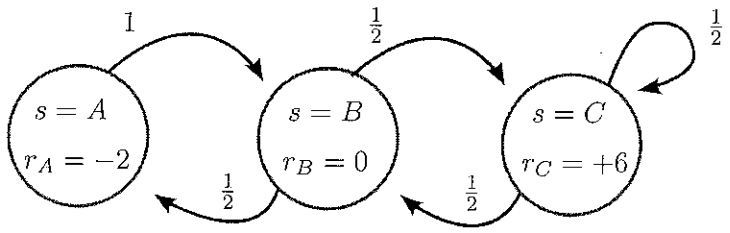


(b) Now execute hierarchical agglomerative clustering on the data points, but use "complete linkage" (maximum distance) for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.



Problem 8: (8 points) Markov models

Consider the Markov model shown here:



$\Pr[A \to B] = 1.0$
$\Pr[B \to A] = \frac{1}{2}$
$\Pr[B \to C] = \frac{1}{2}$
$\Pr[C \to B] = \frac{1}{2}$
$\Pr[C \to C] = \frac{1}{2}$

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 $\gamma = \frac{1}{2}$.

(a) Compute $J^1(s)$, the expected discounted reward for state sequences of length 1 starting in each state s.

$J^1 =$

A	B	C
-2	0	6

(b) Compute $J^2(s)$, the expected discounted reward for state sequences of length 2 starting is each state s.

$J^2 =$

A	B	C
$-2 \cdot 1/2 \cdot 1 \cdot 0$	$0 + 1/2 \cdot 1/2 \cdot (-2)$	$6 + 1/2 \cdot 1/2 \cdot 0$
$= -2$	$+ 1/2 \cdot 1/2 \, (6)$	$+ 1/2 \cdot 1/2 \cdot 6$
	$= 1$	$= 7\,1/2.$

(c) Compute $J^3(s)$, the expected discounted reward for state sequences of length 3 starting is each state s.

$J^3 =$

A	B	C
$-2 + 1/2 \cdot 1 \, (1)$	$0 + 1/2 \cdot 1/2 \, (-2)$	$6 + 1/2 \cdot 1/2 \cdot (1)$
$= -1\,1/2$	$+ 1/2 \cdot 1/2 \, (7\,1/2)$	$+ 1/2 \cdot 1/2 \, (7\,1/2)$
	$= \frac{15}{8} - 1/2$	$= 6 + 1/4 + 15/8$
	$= 11/8$	$= 8\,1/8.$

CS178 Final Exam

Machine Learning & Data Mining: Winter 2016

Thursday March 17th, 2016

Your name: SOLUTIONS

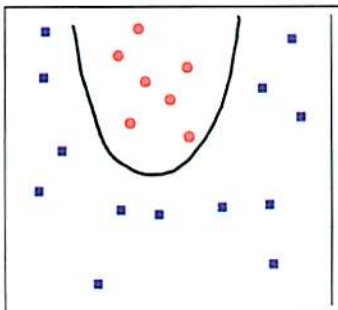
Your ID Number and UCINetID
(e.g., 12345678 / myname@uci.edu): 31415926 / pcreater@uci.

Your seat (row and number): Lecture

- 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 to come over.
- You may use one sheet of your own, handwritten notes for reference, and a calculator.
- Turn in any scratch paper with your exam

Problem 1: (8 points) Separability & Classifiers

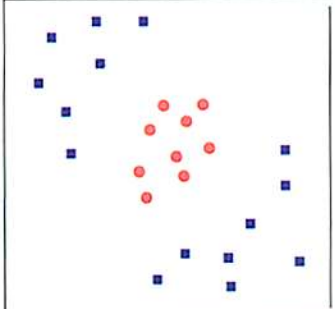
For each of the following examples of training data and classifiers, state whether there exists a set of parameters that can separate the data and justify your answer briefly (~1 sentence).



Linear classifier with quadratic features:

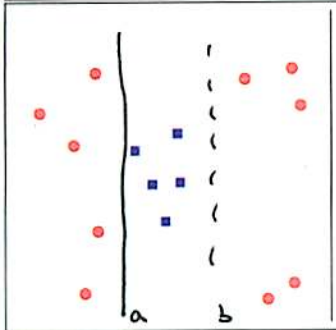
Yes - sketched boundary is

$x_2 = ax_1^2 + bx_1 + c$



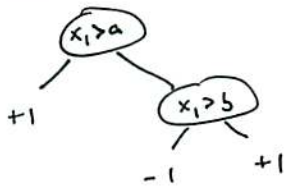
Depth-two decision tree:

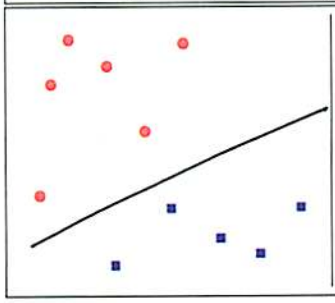
No - no 2 level axis-aligned split.



Depth-two decision tree:

Yes, eg:



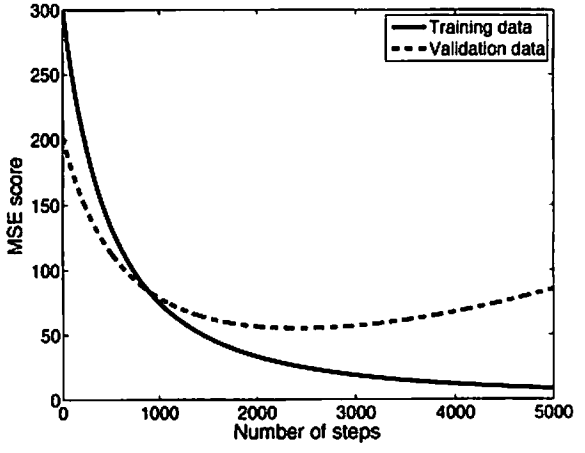


Linear perceptron classifier:

Yes; linear decision boundary.

Problem 2: (9 points) Training & Test Error

Consider the following plot, which shows the training set error and the validation test set error for a neural network model as it is trained, i.e., the horizontal axis indicates the number of iterations of training (gradient steps). Note that the training error decreases monotonically, while the test error does not.



(a) Explain what is happening and why; suggest a possible solution.

Overfitting - the model is becoming too tuned to the training data, degrading validation performance.

Use early stopping, regularization, or some other form of complexity control.

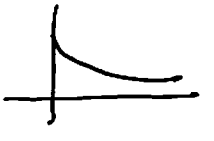
Now suppose that we were to re-train the model with 10 times as much data, while keeping all other aspects (initialization, etc.) the same.

(b) Would you expect the training curve to be different? If so, sketch how it might change.

No / Similar - maybe slightly higher, esp. at the far right.

(c) Would you expect the validation (test) curve to be different? If so sketch how it might change.

Yes - flatter, probably no or less upturn at the far right



Problem 3: (12 points) Decision Trees

Consider the table of measured data given at right. (Note that some data points are repeated.) 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	1	1
0	1	0	1
1	1	1	1
1	1	1	1
0	0	0	0
0	0	0	0
1	1	0	0

(a) What is the entropy of y? $p(y) = 4/7$

$H = 4/7 \log 7/4 + 3/7 \log 7/3 \approx .985$ bits

(b) Which variable would you split first? Justify your answer.

X_3 clearly has lowest entropy after split (highest info gain)

X_1: 0 ⇒ 1100
1 ⇒ 110

X_2: 0 ⇒ 100
1 ⇒ 1110

X_3: 0 ⇒ 1000
1 ⇒ 111

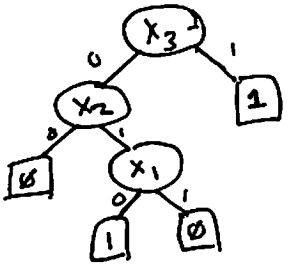
(c) What is the information gain of the variable you selected in part (b)?

$H(4/7) - 4/8\, H(1/4) - 3/7 \cdot 0.$

$\approx .985 - .463 - 0.$

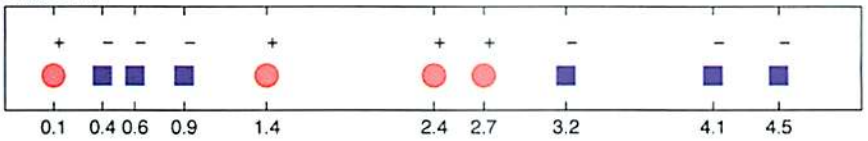
$\approx .522$ bits.

(d) Draw the rest of the decision tree learned on these data.



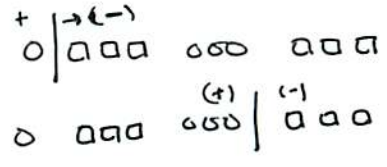
Problem 4: (12 points) Classification in 1D

We observe a collection of training data with one feature, "x" and a class label $y \in \{-, +\}$, shown here; class + is indicated by circles and − by squares, and also labeled with text for redundancy. Answer each of the following questions. Express error rates as the fraction of data points incorrectly classified.



(a) What is the best training error rate we can achieve on these data from a linear classifier on the original input features ($f(x) = \text{sign}(ax + b)$)? Explain briefly (sketch + 1-2 sentences): how it is achieved.

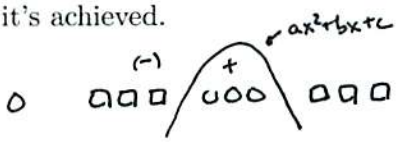
3/10



set the decision boundary, $ax+b=0$, at $x=.3$ or $x=3$, for example.

(b) What is the best training error we can achieve from a linear classifier with quadratic features, e.g., $f(x) = \text{sign}(ax^2 + bx + c)$? Explain briefly how it's achieved.

1/10



$ax^2+bx+c = -(x-1)(x-3)$.

(for example)

(c) What is the best training error we can achieve from a decision tree classifier? Explain briefly how it's achieved.

0/10 keep splitting until all the training data are correct.

(d) What is the best training error we can achieve from a two-layer neural network (multi-layer perceptron) with input features "x" and "1"? Explain briefly how it's achieved (e.g., # of hidden nodes & what they look like).

0/10

Use say 3 hidden nodes, activating at $x=.3$, $x=1$, and $x=3$.

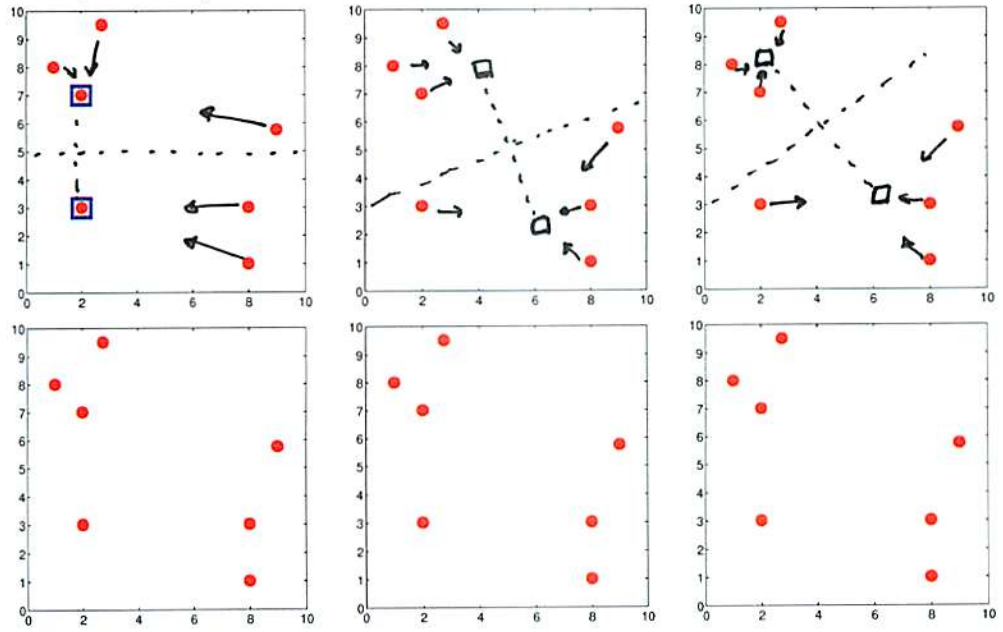
Then, a linear combination of these values can separate the data.

Problem 5: (12 points) Clustering

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

k-means

(a) Starting from the two cluster centers indicated by squares, perform k-means clustering on the data points. In each panel, indicate (somehow) the data assignment, and in the next panel show the new cluster centers. Stop when converged, or after 6 steps (3 iterations), whichever is first. It may be helpful to recall from our nearest-neighbor classifier that the set of points nearer to A than B is separated by a line.



stop.

(b) Write down the cost function optimized by the k-means algorithm, explaining your notation.

$$J = \sum_i \| x^{(i)} - \mu_{z_i} \|_2^2$$

$\|v\|_2^2 = \sum_j v_j^2$ is Euclidean length squared

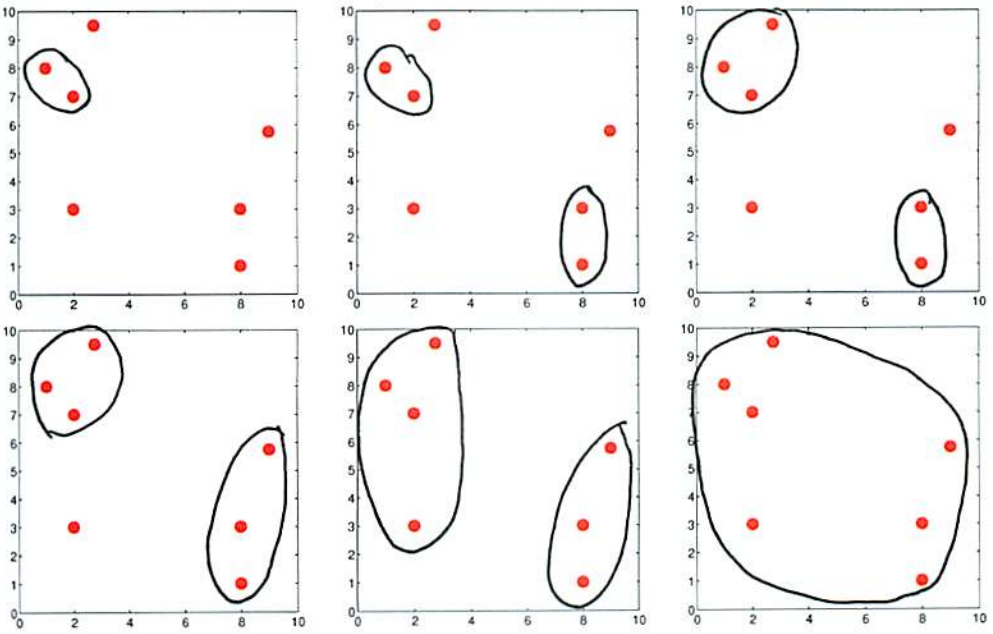
$x^{(i)}$ = data point i.

z_i = cluster assignment of $x^{(i)}$

μ_c = cluster center of cluster c.

Linkage

(a) Now execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "complete linkage" (maximum distance) for the cluster scores. Stop when the algorithm would terminate, or after 6 steps, whichever is first. Show each step separately in a panel.



(b) What is the algorithmic (computational) complexity of the hierarchical clustering algorithm? Briefly justify your answer.

$O(M^2 \log M)$

Compute $O(M^2)$ cluster distances for the original M points.

Maintain sorted $\Rightarrow$ $O(M^2 \log(M^2)) = O(M^2 \log M)$.

Each iteration i (M times):

merge closest cluster pair (constant time)

update $(M-i)$ cluster distances in sorted structure $\Rightarrow$ $O((M-i) \log M)$

$\Rightarrow$ total $O(M^2 \log M)$.

Problem 6: (9 points) Bayes Classifiers and Naïve Bayes

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 $y = 0$.

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

(a) Write down the probabilities necessary for a naïve Bayes classifier:

$p(y=1) = 4/7$ $\quad p(y=0) = 1 - p(y=1) = 3/7$

$p(x_1=1 \mid y=1) = 3/4$ $\quad p(x_1=1 \mid y=0) = 1/3$

$p(x_2=1 \mid y=1) = 1/2$ $\quad p(x_2=1 \mid y=0) = 2/3$

$p(x_3=1 \mid y=1) = 1/2$ $\quad p(x_3=1 \mid y=0) = 1/3$

(b) Using your naïve Bayes model, what value of y is predicted given observation $(x_1, x_2, x_3) = (000)$?

$3/7 \cdot 2/3 \cdot 1/3 \cdot 2/3$ vs $4/7 \cdot 1/4 \cdot 1/2 \cdot 1/2$

$4/9$ vs $1/4$ $\Rightarrow$ predict $y=0$.

(c) What is the class probability $p(y = 1 | x_1 = 0, x_2 = 1, x_3 = 1)$?

$$p(y=1 \mid 011) = \frac{4/7 \cdot 1/4 \cdot 1/2 \cdot 1/2}{4/7 \cdot 1/4 \cdot 1/2 \cdot 1/2 + 3/7 \cdot 2/3 \cdot 2/3 \cdot 1/3} = \frac{4 \cdot 9 \cdot 7}{4 \cdot 27 + 3 \cdot 4 \cdot 4 \cdot 4} = \frac{9}{9+16} = 9/25.$$

Problem 7: (9 points) VC Dimension

Consider the following classifiers $f(x)$, defined on data with two real-valued features $x = (x_1, x_2)$ and predicting a binary class $y \in \{-1, +1\}$. Answer the following questions about their VC dimension, by showing that it is at least as large as the value you give.

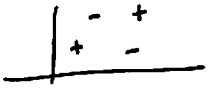
(a) What is the VC dimension of the linear classifier,

$$f(x) = \text{sign}(a + bx_1 + cx_2),$$

with parameters (a, b, c)?

3 - perceptron in 2 features has VC dim 2+1

Recall: can shatter: but cannot separate

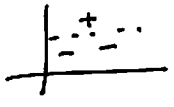


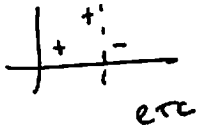
(b) What is the VC dimension of a decision stump,

$$f(x) = a \texttt{ if } x_i < t \texttt{ else } b\,,$$

with parameters (a, b, t, i)?

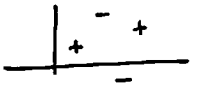
3 -





etc.

Not required, but again, cannot shatter



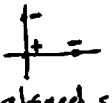
No single axis split.

(c) Suppose that the data points were forced to have *binary valued* features, e.g., $x_i \in \{0, 1\}$, rather than real values. Would this change your answer for (a) or (b)?

For (a) - No. Just place the 3 points at

2 still shatters.

For (b) - Yes. For point placement like ↗,

we cannot separate 

with a single, axis aligned split.

CS273A Final Exam

Introduction to Machine Learning: Winter 2019

Thursday, March 21st, 2019

Your name: Solutions

Row/Seat Number: Lecture

Your ID #(e.g., 123456789) ~~A20~~ 31415926 5

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

- 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.

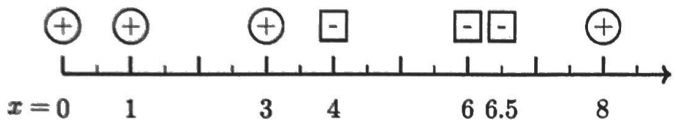
Problems

Total, *(79 points.)*

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.)*

0

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

✓ ✓ ✗ ✗ ✓ ✓ ✗

3/7

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

✓ ✓ ✗ ✓ ✓ ✓ ✗

2/7

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

✓ ✓ ✗ ✓ ✓ ✓ ✗

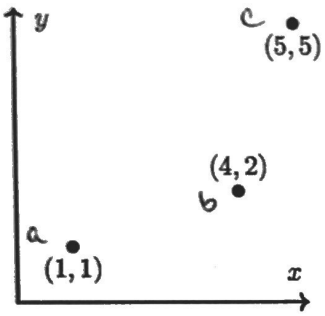
2/7

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.)*



(MSE)

leave out

a ⇒ predict 7/2 ⇒ $(5/2)^2$.

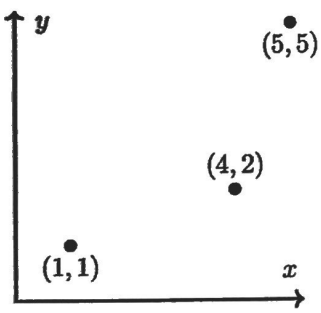
b ⇒ " 3 ⇒ $(1)^2$

c ⇒ " 3/2 ⇒ $(7/2)^2$

$MSE = 1/3\left[(5/2)^2 + (1)^2 + (7/2)^2\right] = 1/3\left[\frac{25}{4} + \frac{4}{4} + \frac{49}{4}\right]$

$= \frac{78}{12} = 6.$

(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.)*



(MSE)

leave out

a ⇒ predict −7 ⇒ $(8)^2$

b ⇒ " 4 ⇒ $(2)^2$

c ⇒ " 7/3 ⇒ $(8/3)^2$

$MSE = 1/3\left[(8)^2 + (2)^2 + (8/3)^2\right] = \frac{676}{27} = 25\ 1/27.$

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) [d] [a] [c] [b] (Most Complex)

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):

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

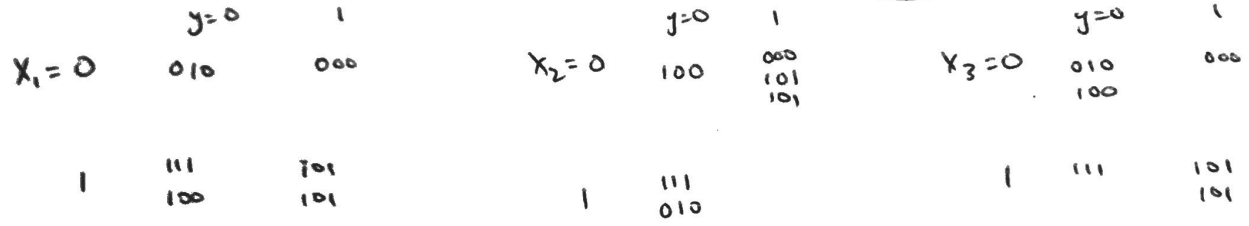
$\log_2(5) = 2.32 \quad \log_2(6) = 2.59 \quad \log_2(7) = 2.81 \quad \log_2(8) = 3$

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.)*

$p(y=1) = 1/2 \Rightarrow H(1/2) = 1$ bit.

(2) Which variable would you split first? Justify your answer. *(2 points.)* X_2



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

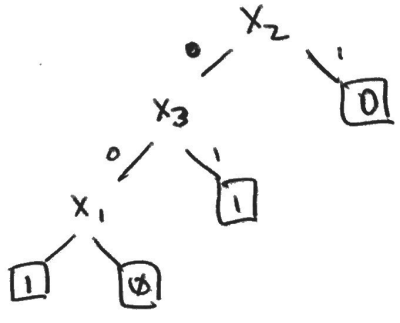
$IG = 1 - \left[1/3\ H(\$) + 2/3\ H(3/4) \right]$

$= 1 - 2/3 \left[3/4 \log 4/3 + 1/4 \log 4 \right]$

($\log 4/3 \approx .41$, $\log 4 = 2$)

$\approx 1 - .64 \approx .36$ bits.

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



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.)*

$p(y=0)$: 1/2 $p(y=1)$: 1/2

$p(x_1=1|y=0)$: 3/4 $p(x_1=1|y=1)$: 1/4

$p(x_1=0|y=0)$: 1/4 $p(x_1=0|y=1)$: 3/4

$p(x_2=1|y=0)$: 1/4 $p(x_2=1|y=1)$: 3/4

$p(x_2=0|y=0)$: 3/4 $p(x_2=0|y=1)$: 1/4

$p(x_3=1|y=0)$: 1/2 $p(x_3=1|y=1)$: 1/2

$p(x_3=0|y=0)$: 1/2 $p(x_3=0|y=1)$: 1/2

(2) **Using your naïve Bayes model,** compute: *(3 points.)*

$p(y=1|x_1=0,x_2=1,x_3=1)$: 1/10 $p(y=0|x_1=0,x_2=1,x_3=1)$: 9/10

$$p(y=1|011) = \frac{1/2 \cdot 1/4 \cdot 1/4 \cdot 1/2}{1/2 \cdot 1/4 \cdot 1/4 \cdot 1/2 + 1/2 \cdot 3/4 \cdot 3/4 \cdot 1/2} = \frac{1}{1+9} = 1/10$$

(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.)*

$p(y=1|011) = 1$

$p(y=0|011) = 0$

by inspection - two examples w/ $y=1$, no examples w/ $y=0$.

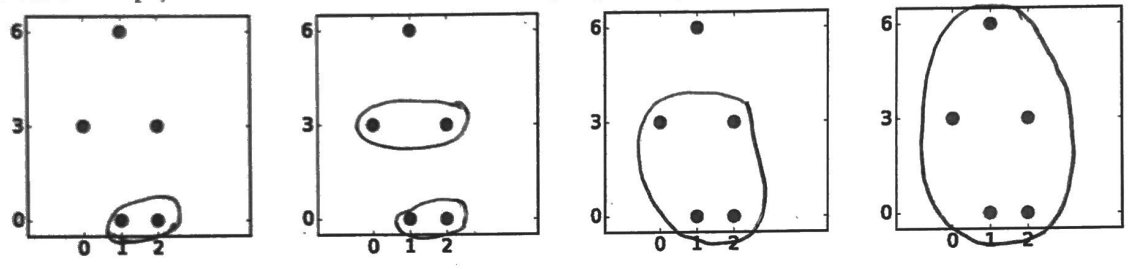
Name: ID#:

Clustering, *(10 points.)*

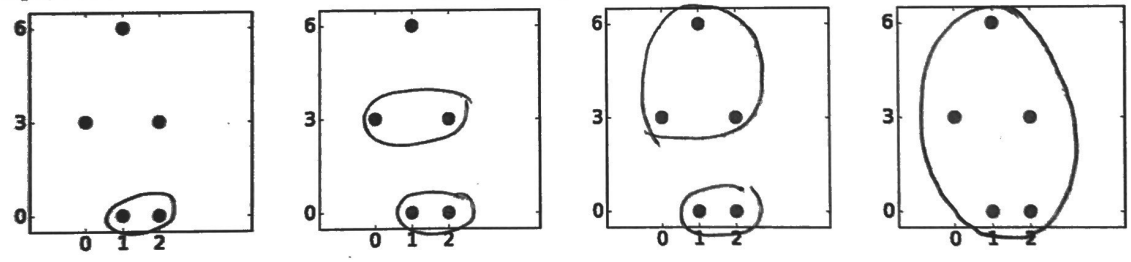
Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data.

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.

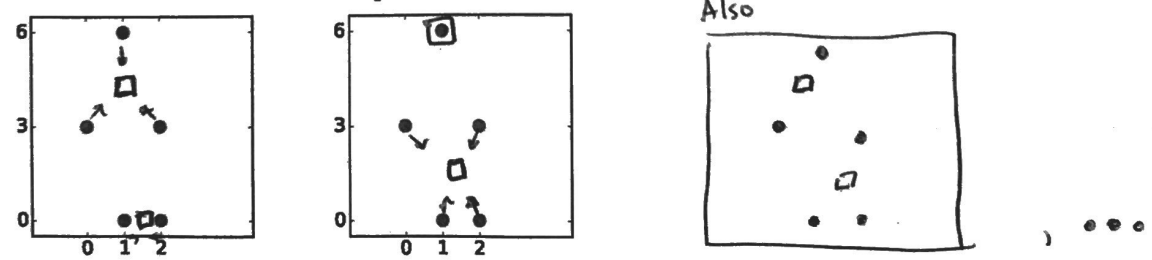


(b) Now repeat your agglomerative clustering algorithm, this time using "complete linkage" (maximum distance) for the cluster scores. Stop when the algorithm would terminate, or after 4 steps, whichever is first. Show each step separately in a panel.



k-means

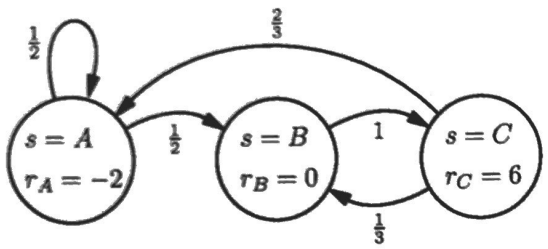
In the panels below, sketch two different clustering solutions that could be found by (would be converged solutions of) k-means, with $k = 2$. Show the final (converged) grouping of data (assignments) and final cluster locations. (You do not have to show the process, only the final clustering.) This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



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.)*

	A	B	C
$J_1(s)$	−2	0	6

(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.)*

	A	B	C
$J_2(s) =$	$-2 + \frac{1}{2}\cdot\frac{1}{2} J_1(A) + \frac{1}{2}\cdot\frac{1}{2} J_1(B)$	$0 + \frac{1}{2}\cdot 1 \cdot J_1(C)$	$6 + \frac{1}{2}\cdot\frac{2}{3}\cdot J_1(A) + \frac{1}{2}\cdot\frac{1}{3} J_1(B)$
$=$	$-2\frac{1}{2}$	3	$5\frac{1}{3}$

(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.)*

	A	B	C
$J^3(s) =$	$-2 + \frac{1}{2}\cdot\frac{1}{2} J_2(A) + \frac{1}{2}\cdot\frac{1}{2} J_2(B)$	$0 + \frac{1}{2}\cdot 1 \cdot J_2(C)$	$6 + \frac{1}{2}\cdot\frac{2}{3} J_2(A) + \frac{1}{2}\cdot\frac{1}{3} J_2(B)$
$=$	$-2\frac{1}{8}$	$2\frac{2}{3}$	$5\frac{2}{3}$

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

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

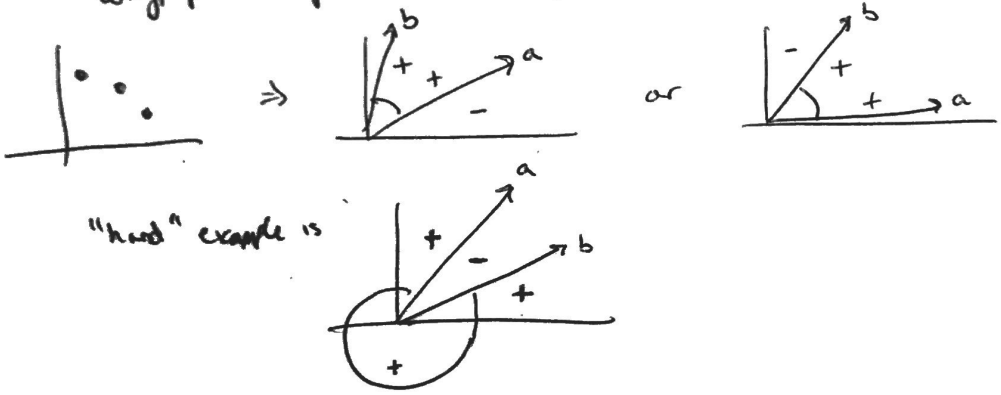
where θ_a, θ_b are parameters to be learned from the data.

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

3

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

WLOG, put the point in the 1st quadrant



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

An impossible pattern is

(or any position around the circle).

We can only transition once & back; "+" region must form a contiguous slice of the circle.

CS273A Final Exam

Introduction to Machine Learning: Winter 2020

Tuesday March 17th, 2020

Your name:

SOLUTIONS

Your ID #(e.g., 123456789)

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

- Due to the ongoing health emergency, this exam is **take home** and may be submitted in person or on Canvas (scanned).
- Total time is 2 hours 15 minutes for either in-person delivery (to the classroom) or submission to Canvas.
- READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Please put your name and ID **on every page**.
- Please **write clearly** and **show all your work**.
- If you need clarification on a problem, please post **privately** on Piazza and we will try to answer it.

Problems

Total, *(74 points.)*

Name: _____ ID#: _____

Bayes Classifiers, *(12 points.)*

In this problem you will use Bayes Rule: $p(y|x) = p(x|y)p(y)/p(x)$ to perform classification. Suppose we observe some training data with two binary features x_1, x_2 and a binary class y. After learning the model, you are also given some validation data.

Table 1: Training Data

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

Table 2: Validation Data

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

In the case of any ties, we will prefer to predict class 0.

(1) Give the predictions of a joint Bayes classifier on the validation data. What is the validation error rate? (Put final answers in boxes.) *(4 points.)*

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

Error Rate:

3/4

(2) Give the required probabilities to define a **naïve** Bayes classifier. *(4 points.)*

$p(y=1) = 1/2 = p(y=0)$

$p(x_1=1 \mid y=0) = 1/2$ $\quad$ $p(x_1=1 \mid y=1) = 1/2$

$p(x_2=1 \mid y=0) = 3/4$ $\quad$ $p(x_2=1 \mid y=1) = 1/2$

(3) Give the predictions of a naïve Bayes classifier on the validation data. What is the validation error rate? (Put final answers in boxes.) *(4 points.)*

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

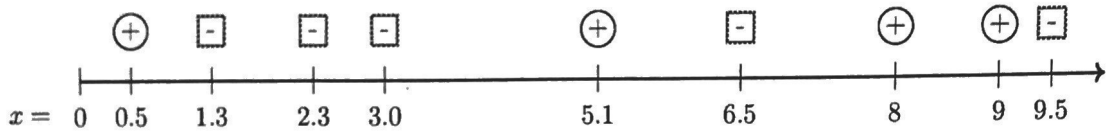
Error Rate:

1/4

Name: _____ ID#: _____

Cross-Validation, *(9 points.)*

Consider the following dataset with *nine* 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 **leave-one-out** cross-validation error rate of a 1-nearest neighbor classifier. *(3 points.)*

✗ ✗ ✓ ✓ ✗ ✗ ✓ ✗ ✗

2/3

(2) Compute the **leave-one-out** cross-validation error rate of a 3-nearest neighbor classifier. *(3 points.)*

✗ ✓ ✓ ✓ ✗ ✗ ✗ ✗ ✗

2/3

(3) Compute the **leave-one-out** cross-validation error rate of an 8-nearest neighbor classifier. *(3 points.)*

All ⊕ wrong
All ⊖ right

4/9

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	0
1	1	0	1
0	1	0	0
0	1	0	0
0	1	0	0
1	0	0	0
1	1	0	1
1	0	1	1

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

$H(3/8) = -3/8 \log 3/8 - 5/8 \log 5/8 \approx .95$ bits

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

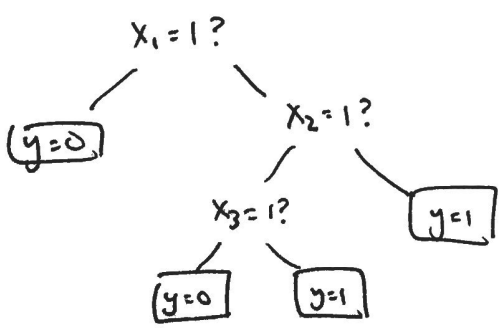
	x_1	x_2	x_3
=0	0000	001	000011
=1	0111	000 11	1

$\Rightarrow$ x_1

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

$IG = (.95) - 4/8\, H(\emptyset) - 4/8\, H(3/4) \approx .5488$

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



If $x_1 = 1$:

110	1	
110	1	
100	0	
101	1	

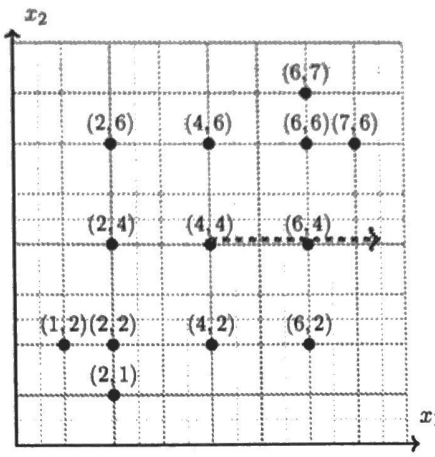
	x_2?	x_3?
=0	01	110
=1	11	1

$\Rightarrow x_2$ better.

Name: _____ ID#: _____

Dimensionality Reduction, *(10 points.)*

(1) For the following points in two dimentions, consider performing linear dimensionality reduction along the given vector (dashed line). What is the reconstruction error, in MSE, when using this vector? *(4 points.)*



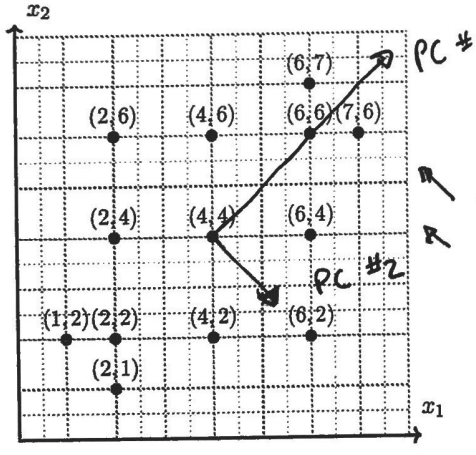
MSE

$\leftarrow 3^2$
$+$
$\leftarrow 2^2 \cdot 4$
$+$
$\leftarrow 0$
$+$
$\leftarrow 2^2 \cdot 4$
$+$
3^2
$/13$

$\Rightarrow 50/13 \approx 3.85.$

(2) On the figure below, draw the directions of the first two principal components. *(2 points.)*

(3) What is the reconstruction error (MSE) of these points when only the first principal component is used to reconstruct each point? *(4 points.)*



MSE

$\sqrt{2} \cdot 1/2 \cdot 2$
$+$
0
$+$
$\sqrt{2} \cdot 2$
$+$
$2\sqrt{2} \cdot 2$
$+$
$\sqrt{2} \cdot 2$
$+$
0
$+$
$\sqrt{2} \cdot 1/2 \cdot 2$

$\approx 14.14/13 \approx 1.087.$

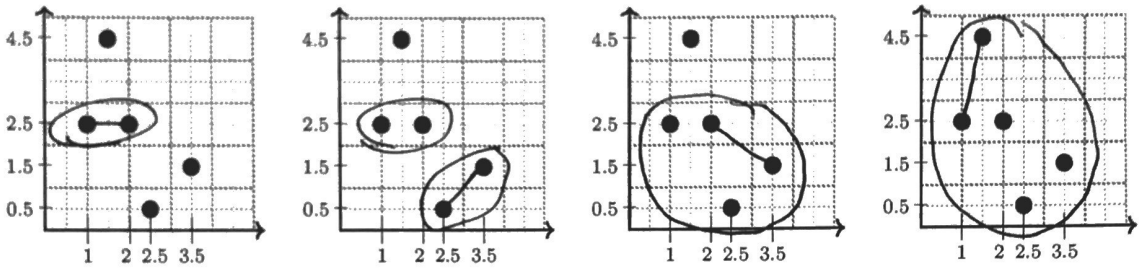
Name: ID#:

Hierarchical Clustering, *(11 points.)*

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

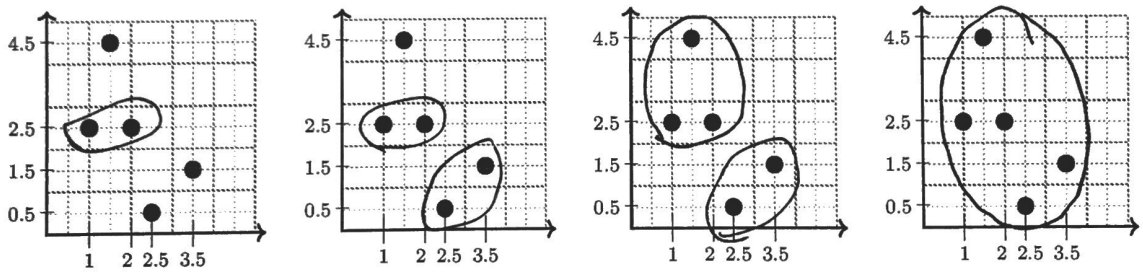
Linkage

(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. *(4 points.)*



line shows closest pair.

(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. *(4 points.)*



(c) What is the (big-O) computational complexity of the hierarchical agglomerative clustering algorithm? Justify your answer **briefly** in 1-2 sentences. *(3 points.)*

$O(n^2)$ - compute all distances

$O(n^2 \log n^2)$ - sort

n iterations:

$O(n)$ distance recomputations + $O(n \log n^2)$ re-insertions

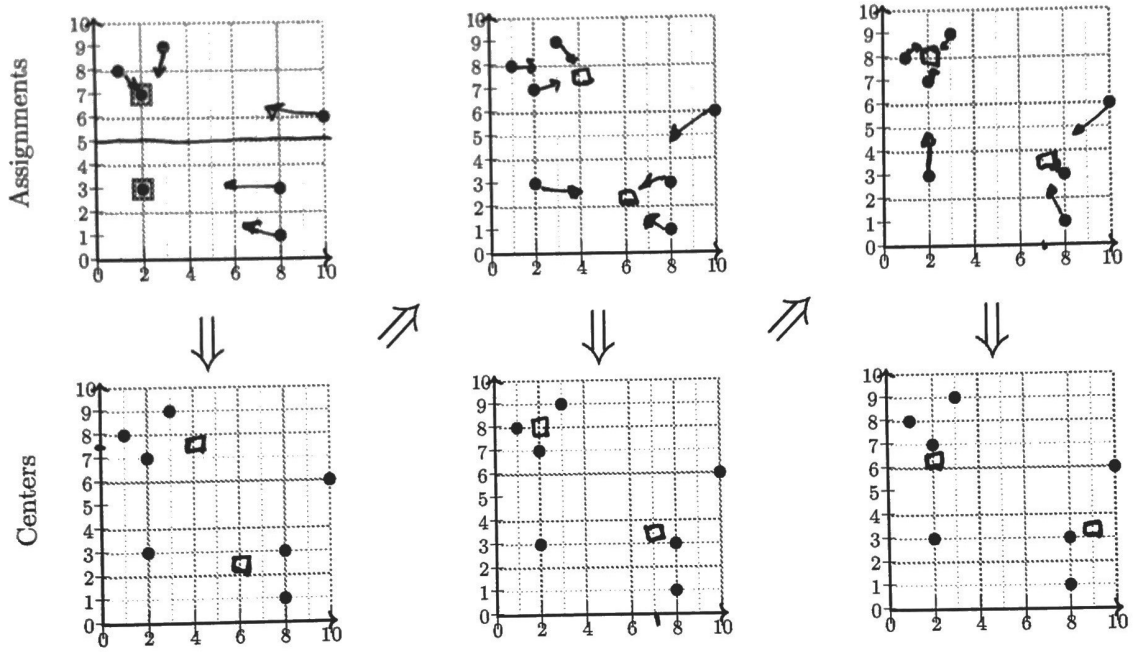
$\Rightarrow O(n^2 + n^2 \log n^2 + n^2 + n^2 \log n^2) = O(n^2 \log n^2) = O(n^2 \log n)$.

Name: _____ ID#: _____

K-Means Clustering, *(10 points.)*

Consider the 2-D data points plotted in each panel. In this problem, we will cluster these data using the k-means algorithm, where each panel is used to show a single step of the algorithm.

(a) Starting from the two cluster centers indicated by squares, perform k-means clustering on the data. In the top panels, indicate the assignment of the data, and then in the panel below show the new cluster centers, so each **pair** of panels shows an iteration of k-means. Stop when converged, or after 6 steps (3 iterations), whichever is first. It may be helpful to recall from our nearest neighbor classifier that the set of points nearer to A than B is separated by a line. *(6 points.)*



(b) Write down the cost function optimized by the k-means algorithm, in terms of the data locations $x^{(i)}$, cluster centers μ_c, and cluster assignments $z^{(i)}$. *(2 points.)*

$$J(\mu, z) = \sum_i \|x^{(i)} - \mu_{z^{(i)}}\|^2 = \sum_i \sum_j \left(x_j^{(i)} - \mu_{z^{(i)}j}\right)^2$$

(c) What is the (big-O) computational complexity of each iteration of k-means (naïve computation), in terms of the data size m and number of clusters k? *(2 points.)*

$O(m \cdot k)$ distance computations to assign

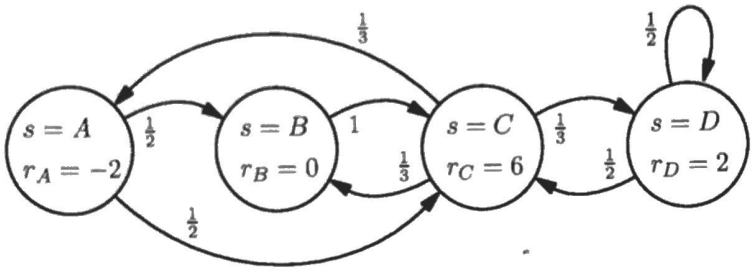
$O(m)$ to compute new means

$\Rightarrow$ $O(mk)$ per iteration.

Name: _____ ID#: _____

Markov Processes, *(12 points.)*

Consider the Markov reward process model shown here:



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

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. We will use dynamic programming to (start) computing the expected discounted sum of rewards. 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. *(4 points.)*

$J^1(A) =$	$J^1(B) =$	$J^1(C) =$	$J^1(D) =$
-2	$\emptyset$	6	2

(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. *(4 points.)*

$J^2(A) =$	$J^2(B) =$	$J^2(C) =$	$J^2(D) =$
$-1/2$	3	6	4
$-2 + 1/2 \cdot 1/2 \cdot \emptyset + 1/2 \cdot 1/2 \cdot 6$	$\emptyset + 1/2 \cdot 1 \cdot 6$	$6 + 1/2 \cdot 1/3 \cdot \emptyset + 1/2 \cdot 1/3 \cdot (-2) + 1/2 \cdot 1/3 \cdot 2$	$2 + 1/2 \cdot 1/2 \cdot 2 + 1/2 \cdot 1/2 \cdot 6$

(3) Compute $J^3(s)$, the expected discounted sum of rewards for state sequences of length 3 (e.g., $[D \to C \to A]$) starting in each state s. *(4 points.)*

$J^3(A) =$	$J^3(B) =$	$J^3(C) =$	$J^3(D) =$
$1/4$	3	$7\ 1/12$	$4\ 1/2$
$-2 + 1/2 \cdot 1/2 \cdot 3 + 1/2 \cdot 1/2 \cdot 6$	$0 + 1/2 \cdot 1 \cdot 6$	$6 + 1/2 \cdot 1/3 \cdot (-1/2) + 1/2 \cdot 1/3 \cdot 3 + 1/2 \cdot 1/3 \cdot 4$	$2 + 1/2 \cdot 1/2 \cdot 4 + 1/2 \cdot 1/2 \cdot 6$

CS 273a Final Exam

Intro to Machine Learning: Fall 2023

Monday December 11th, 2023

Your name: Peter Anteater

Row/Seat Number:

Your ID #(e.g., 123456789) 314159265

UCINetID (e.g.ucinetid@uci.edu) panteater@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**.
- Please write your final answer **in the provided box** where possible.
- You may use **one** sheet containing handwritten notes for reference, and a **basic** calculator; no other electronics allowed.
- If you need clarification on a problem, please raise your hand and wait for the instructor or TA to come over.

Problems

Total, *(74 points.)*

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: ID#:

Problem 1 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:

$\log_2(1) = 0 \quad \log_2(2) = 1 \quad \log_2(3) = 1.59 \quad \log_2(4) = 2$
$\log_2(5) = 2.32 \quad \log_2(6) = 2.59 \quad \log_2(7) = 2.81 \quad \log_2(8) = 3$
$\log_2(9) = 3.17 \quad \log_2(10) = 3.32$
Note also that $\log(a/b) = \log(a) - \log(b)$.

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

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

$H(5/9) = -5/9 \log 5/9 - 4/9 \log 4/9$
$= 5/9 (.85) + 4/9 (1.17)$
$= .47 + .52 = .99$

.99 bits

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

	x_1	x_2	x_3
$x_i = 0$?	001	10	10011
$x_i = 1$?	11100	1001101	1001

$\Rightarrow$ x_1, lowest entropy / highest IG.

x_1

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

$IG = H(5/9) - 1/3 H(1/3) - 2/3 H(1/3)$
$= .99 - .92 = .08$
(.073 by calculator)

.073 ≈ .08

(4) Draw the full decision tree learned on these data. *(4 points.)*

$x_1 = 0$?

0 1 0 0
0 1 0 0
0 1 1 1

$\Rightarrow$ x_3 next

$x_1 = 1$?

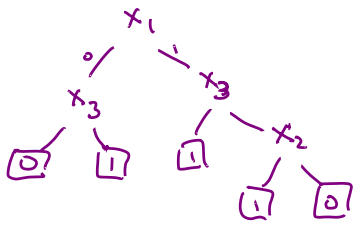
1 1 0 1
1 0 1 1
1 1 0 1
1 1 0 1
1 1 1 0
1 0 1 0

$\Rightarrow$ x_2? 0 $\Rightarrow$ 01, 1 $\Rightarrow$ 1110

x_3? 0 $\Rightarrow$ 111 ✓, 1 $\Rightarrow$ 100

$x_1 = 1, x_3 = 1$?

1011
1010 > tie, predict 1
1110



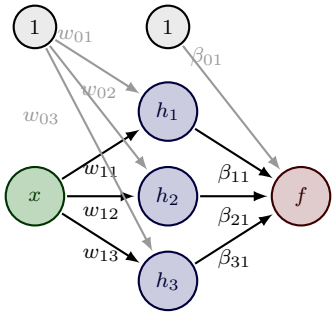
This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: _____ ID#: _____

Problem 2 Neural Networks, *(9 points.)*

Consider a small neural network designed to classify a scalar feature x as one of $y \in \{-1, +1\}$. We have three hidden nodes h_1, h_2, h_3 and a single output node f_1.



You are given the weights W of the hidden layer,

$$W = \begin{bmatrix} w_{01} & w_{11} \\ w_{02} & w_{12} \\ w_{03} & w_{13} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 3 & -2 \\ -3 & 3 \end{bmatrix}$$

and the weights B of the output layer,

$$B = \begin{bmatrix} \beta_{01} & \beta_{11} & \beta_{21} & \beta_{31} \end{bmatrix} = \begin{bmatrix} -1 & 3 & -1 & -1 \end{bmatrix}.$$

(For example, w_{12} is the weight connecting x_1 to h_2; w_{02} is the constant (bias) term for h_2, etc.)

The network uses a ReLU activation function, $a(z) = \max(0, z)$, for the hidden layer, and a logistic sigmoid activation function,

$$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{1}{1 + \frac{1}{\exp(z)}} = \frac{\exp(z)}{\exp(z) + 1},$$

for the output layer (the value of which, as usual, corresponds to the model's probability that the class is $+1$). You may find the following values useful:

$\exp(0) = 1$ $\exp(1) = 2.72$ $\exp(2) = 7.39$ $\exp(3) = 20.09$ $\exp(4) = 54.60$ $\exp(5) = 148.40$

---

(1) What class is predicted by the model given the input $x_1 = 2$? *(3 points.)*

$x_1 = 2 \Rightarrow r_1 = 0 + 2 = 2 \quad h_1 = 2$

$r_2 = 3 - 4 = -1 \Rightarrow h_2 = 0 \Rightarrow s = -1 + 0 - 0 - 3$

$r_3 = -3 + 6 = 3 \quad h_3 = 3 \qquad = 2$

$\sigma(2) = \frac{\exp(2)}{\exp(2)+1} = \frac{7.4}{8.4} = 0.88 > \frac{1}{2}$

$\hat{y} = +1$

(2) What is the model's estimated *probability* $p(y = +1 \mid x_1 = 2)$? *(3 points.)*

0.88

(3) Suppose our input is $x_1 = 0$: what is the probability $p(y = +1 \mid x_1 = 0)$? *(3 points.)*

$x_1 = 0 \Rightarrow r_1 = 0 + 0 = 0 \quad h_1 = 0$

$r_2 = 3 - 0 = 3 \Rightarrow h_2 = 3 \Rightarrow s = -1 + 0 - 3 - 0$

$r_3 = -3 + 0 = -3 \quad h_3 = 0 \qquad = -4$

$\sigma(-4) = \frac{1}{1+\exp(4)} = \frac{1}{55.6} \approx .018$

.018

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: ID#:

Problem 3 True/False, *(10 points.)*

For each of the scenarios below, circle one of "true" or "false" to indicate whether you agree with the statement.

True or **false**: Changing our classifier from a nearest centroid classifier to a Gaussian Bayes classifier is likely to reduce our model's bias.

True or **false**: When training a 1-nearest neighbor classifier, if we decide to double the amount of data available to the learner, we would expect the bias to decrease.

True or **false**: When learning a decision tree model, if we restrict our learner to only use odd-numbered features, we will reduce our variance.

True or **false**: In a random forest model, we use the best-fitting tree in our ensemble for prediction.

True or **false**: In a random forest model, we usually learn using a random subset of features at each node in order to speed up the learning process (reduce computation).

True or **false**: If, when training a random forest model, we decide to use the same subset of features across all levels of a given tree (i.e., the entire tree uses only those features), we will increase our model's bias.

True or **false**: When clustering using k-means, we can select the number of clusters k using a hold-out (validation) data set.

True or **false**: Computing the expected discounted sum of rewards in a Markov Reward Process can be solved using either matrix algebra or dynamic programming.

True or **false**: The larger the value of gamma, the less far-sighted (less interested in future rewards) the RL agent becomes.

True or **false**: Once the optimal state-action value function is computed, the reinforcement learning problem can be considered solved.

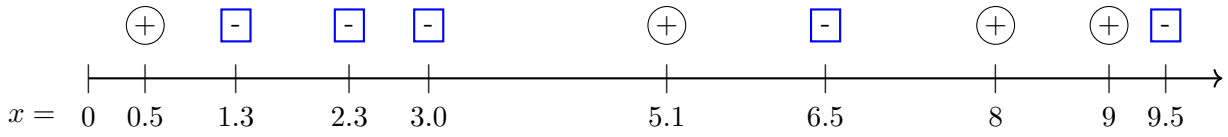
This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: ID#:

Problem 4 K-Nearest Neighbors, *(12 points.)*

Consider the following dataset with *nine* 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. *(3 points.)*

∅

(2) Compute the **leave-one-out** cross-validation error of 1-nearest neighbor classifier. *(3 points.)*

✗ ✗ ✓ ✓ ✗ ✗ ✓ ✗ ✗

6/9

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

✗ ✓ ✓ ✓ ✗ ✓ ✓ ✗ ✓

3/9

(4) Compute the **leave-one-out** cross-validation error of 2-nearest neighbor classifier. *(3 points.)*

✗ ✓ ✓ ✓ ✗ ✗ ✗ ✗ ✗

6/9

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

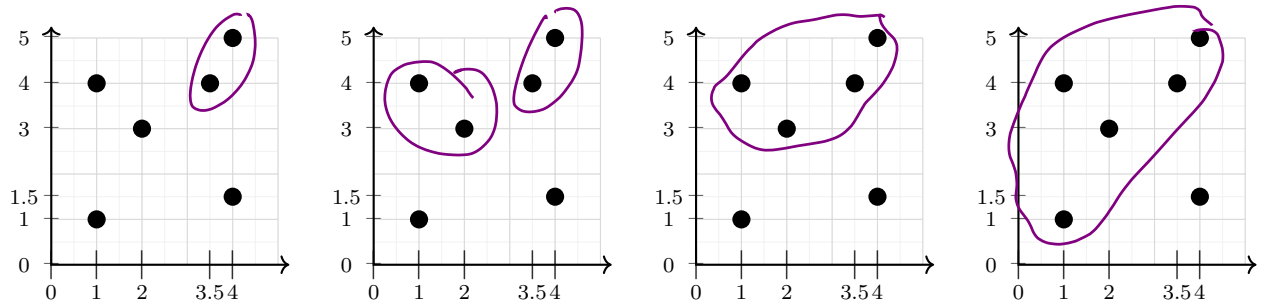
Name: ID#:

Problem 5 Hierarchical Clustering, *(12 points.)*

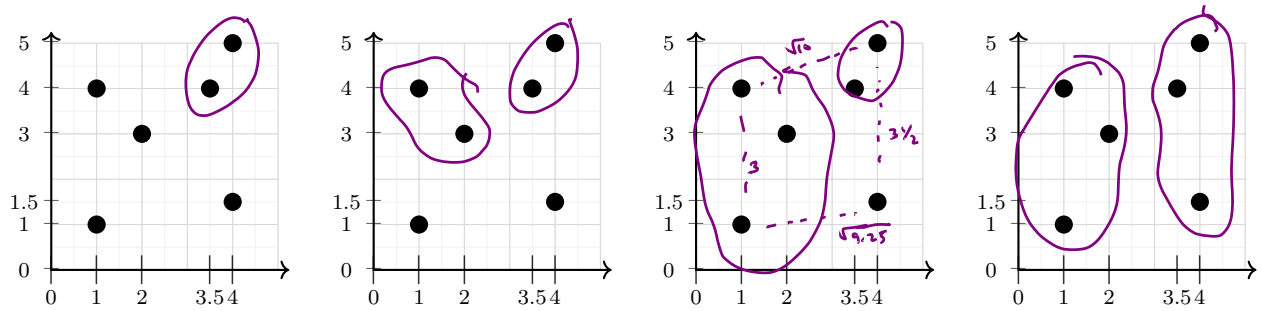
Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data.

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. *(6 points.)*



(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. *(6 points.)*



This page is intentionally blank.

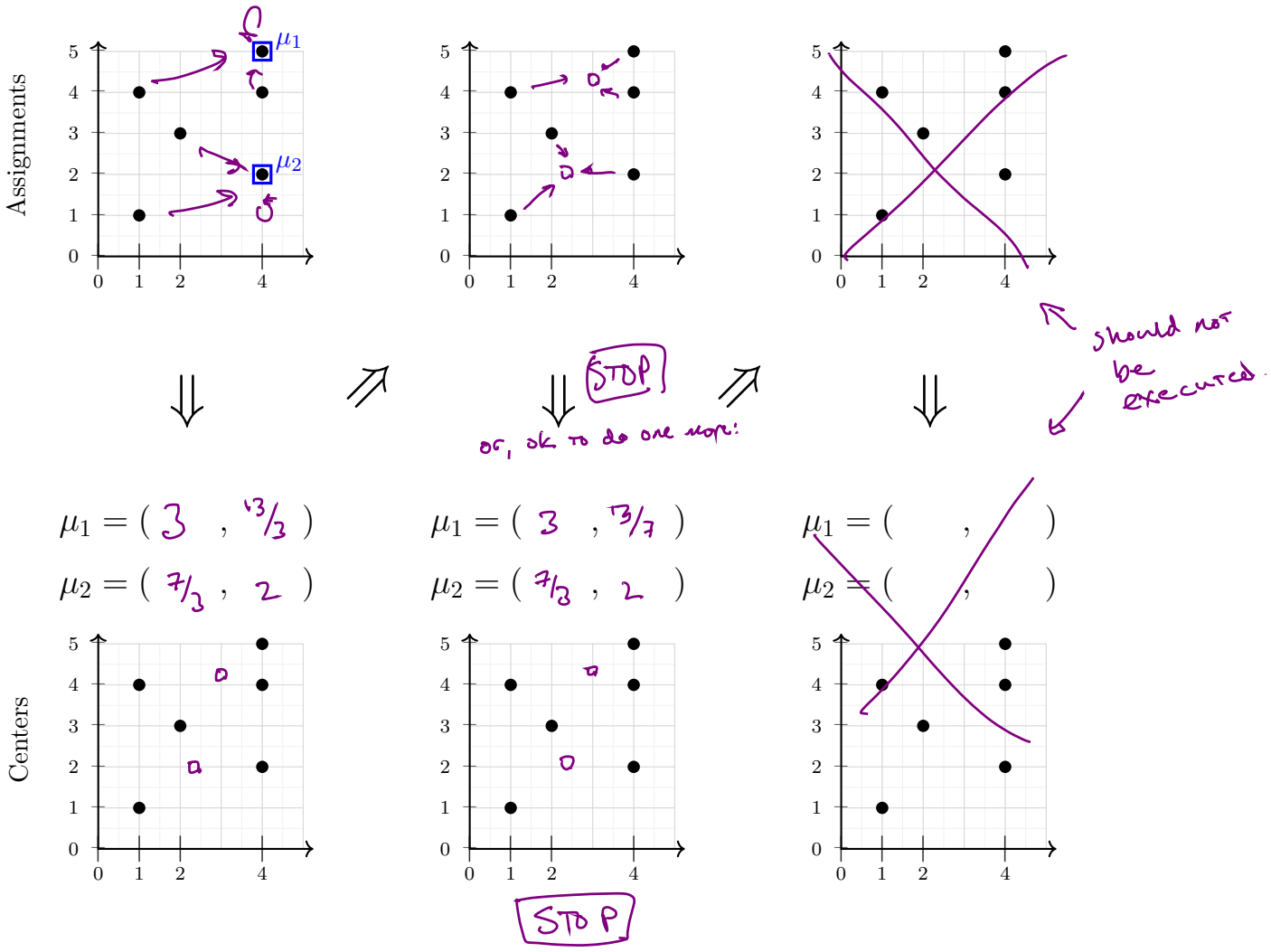
While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: ID#:

Problem 6 K-Means Clustering, *(9 points.)*

Consider the 2-D data points plotted in each panel. In this problem, we will cluster these data using the k-means algorithm, where each panel is used to show a single step of the algorithm.

Starting from the two cluster centers indicated by squares, perform k-means clustering on the data. In the top panels, indicate the assignment of the data, and then in the panel below give the values of the updated cluster centers and sketch their location in the plot, so each **column** of panels shows an iteration of k-means. Stop when converged, or after 6 steps (3 iterations), whichever is first. In the case of any ties, we will prefer to assign to cluster 1. It may be helpful to recall from our nearest neighbor classifier that the set of points nearer to A than B is separated by a line.



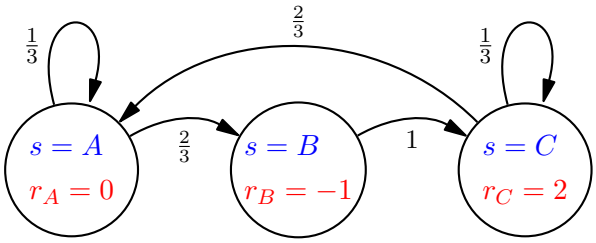
This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

Name: ID#:

Problem 7 Markov Processes, *(12 points.)*

Consider the Markov reward process model shown here:



$\Pr[A \to A] = 0.33$
$\Pr[A \to B] = 0.67$
$\Pr[B \to C] = 1.0$
$\Pr[C \to A] = 0.67$
$\Pr[C \to C] = 0.33$

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. We will use dynamic programming to (start) computing the expected discounted sum of rewards. 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. *(4 points.)*

$J^1(A) =$ 0

$J^1(B) =$ −1

$J^1(C) =$ 2

(2) Compute $J^2(s)$, the expected discounted sum of rewards for state sequences of length 2 (e.g., $[C \to A]$) starting in each state s. *(4 points.)*

$J^2(A) =$ $-1/3 \approx -.333$

$0 + \frac{1}{2}\left(\frac{1}{3}\cdot 0 + \frac{2}{3}\cdot -1\right) = -\frac{1}{3}$

$J^2(B) =$ 0

$-1 + \frac{1}{2}(1 \cdot 2) = 0$

$J^2(C) =$ $7/3 \approx 2.33$

$2 + \frac{1}{2}\left(\frac{1}{3}\cdot 2 + \frac{2}{3}\cdot 0\right) = \frac{7}{3}$

(3) Compute $J^3(s)$, the expected discounted sum of rewards for state sequences of length 3 (e.g., $[C \to A \to A]$) starting in each state s. *(4 points.)*

$J^3(A) =$ $-2/9 \approx -.222$

$0 + \frac{1}{2}\left(\frac{1}{3}\cdot -\frac{1}{3} + \frac{2}{3}\cdot 0\right) = -\frac{2}{9}$

$J^3(B) =$ $1/6 \approx .16$

$-1 + \frac{1}{2}\left(1 \cdot \frac{7}{3}\right) = \frac{1}{6}$

$J^3(C) =$ $41/18 \approx 2.277$

$2 + \frac{1}{2}\left(\frac{1}{3}\cdot \frac{7}{3} + \frac{2}{3}\left(-\frac{1}{3}\right)\right) = \frac{41}{18}$

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

This page is intentionally blank.
It may be used for scratch work or to provide additional space for solutions if necessary.

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.

This page is intentionally blank.
It may be used for scratch work or to provide additional space for solutions if necessary.

This page is intentionally blank.

While you may use it as scratch paper, it may not be scanned for grading, so please do not include final solutions.